TESIS - SS142501

# KLASIFIKASI DATA BERDIMENSI TINGGI DENGAN METODE *ENSEMBLE* BERBASIS REGRESI LOGISTIK DALAM PERMASALAHAN *DRUG DISCOVERY*

T. DWI ARY WIDHIANINGSIH
NRP. 06211650010024

DOSEN PEMBIMBING
Dr.rer.pol. Heri Kuswanto, S.Si., M.Si.
Dr.rer.pol. Dedy Dwi Prastyo, S.Si., M.Si.

PROGRAM MAGISTER
DEPARTEMEN STATISTIKA
FAKULTAS MATEMATIKA, KOMPUTASI DAN SAINS DATA
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2018

**TESIS - SS142501**

# KLASIFIKASI DATA BERDIMENSI TINGGI DENGAN METODE *ENSEMBLE* BERBASIS REGRESI LOGISTIK DALAM PERMASALAHAN *DRUG DISCOVERY*

**T. DWI ARY WIDHIANINGSIH**
**NRP. 06211650010024**

**DOSEN PEMBIMBING**
**Dr.rer.pol. Heri Kuswanto, S.Si., M.Si.**
**Dr.rer.pol. Dedy Dwi Prastyo, S.Si., M.Si.**

**PROGRAM MAGISTER**
**DEPARTEMEN STATISTIKA**
**FAKULTAS MATEMATIKA, KOMPUTASI DAN SAINS DATA**
**INSTITUT TEKNOLOGI SEPULUH NOPEMBER**
**SURABAYA**
**2018**

THESIS - SS142501

# *CLASSIFICATION OF HIGH DIMENSIONAL DATA USING ENSEMBLE BASED METHOD OF LOGISTIC REGRESSION APPLIED TO DRUG DISCOVERY*

**T. DWI ARY WIDHIANINGSIH**
**NRP. 06211650010024**

**SUPERVISORS**
**Dr.rer.pol. Heri Kuswanto, S.Si., M.Si.**
**Dr.rer.pol. Dedy Dwi Prastyo, S.Si., M.Si.**

**MASTER PROGRAM**
**DEPARTMENT OF STATISTICS**
**FACULTY OF MATHEMATICS, COMPUTING AND DATA SCIENCE**
**INSTITUT TEKNOLOGI SEPULUH NOPEMBER**
**SURABAYA**
**2018**

**KLASIFIKASI DATA BERDIMENSI TINGGI DENGAN METODE *ENSEMBLE* BERBASIS REGRESI LOGISTIK DALAM PERMASALAHAN *DRUG DISCOVERY***

**Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar**
**Magister Sains (M.Si)**
**di**
**Institut Teknologi Sepuluh Nopember**

**oleh :**
**T. DWI ARY WIDHIANINGSIH**
**NRP. 06211650010024**

**Tanggal Ujian : 11 Juli 2018**
**Periode Wisuda : September 2018**

**Disetujui oleh:**

1. **Dr.rer.pol. Heri Kuswanto, S.Si., M.Si.** **(Pembimbing I)**
   **NIP: 19820326 200312 1 004**

2. **Dr.rer.pol. Dedy Dwi Prastyo, S.Si., M.Si.** **(Pembimbing II)**
   **NIP: 19831204 200812 1 002**

3. **Santi Puteri Rahayu, M.Si., Ph.D.** **(Penguji)**
   **NIP: 19750115 199903 2 003**

4. **Santi Wulan Purnami, M.Si., Ph.D.** **(Penguji)**
   **NIP: 19720923 199803 2 001**

**Dekan**
**Fakultas Matematika, Komputasi dan Sains Data**
**Institut Teknologi Sepuluh Nopember**

**Prof. Dr. Basuki Widodo, M.Sc.**
**NIP: 19650605 198903 1 002**

# KLASIFIKASI DATA BERDIMENSI TINGGI DENGAN METODE *ENSEMBLE* BERBASIS REGRESI LOGISTIK DALAM PERMASALAHAN *DRUG DISCOVERY*


Nama Mahasiswa : T. Dwi Ary Widhianingsih
NRP : 06211650010024
Dosen Pembimbing : Dr.rer.pol. Heri Kuswanto, S.Si, M.Si
Dr.rer.pol. Dedy Dwi Prastyo, S.Si, M.Si


## ABSTRAK


Regresi logistik biner merupakan salah satu metode yang sering digunakan dalam analisis klasifikasi. Akan tetapi penerapannya pada data berdimensi tinggi mengalami banyak kendala, seperti adanya multikolinearitas, *overfitting*, dan kompleksitas dalam komputasinya. Oleh karena itu, pengembangan banyak dilakukan untuk mengatasi permasalahan tersebut. Salah satunya adalah dengan menerapkan konsep *ensemble*, seperti *Logistic Regression Ensemble* (Lorens) dan *Ensemble Logistic Regression* (ELR). Lorens dibangun dengan konsep *ensemble* dengan pembagian variabel ke dalam beberapa *subspace* secara random dan saling bebas, kemudian hasil analisisnya digabungkan dengan perhitungan rata-rata atau *majority voting*. ELR dibuat menggunakan konsep *ensemble* yang melakukan pengulangan pada proses pembuatan modelnya hingga kriteria konvergen yang didefinisikan tercapai. Karena berawal dari konsep pengembangan dan dengan tujuan yang sama, maka pada penelitian ini dilakukan perbandingan antara kedua metode tersebut, khususnya jika diterapkan untuk data berdimensi tinggi, yaitu dengan studi kasus *drug discovery*. Dataset yang digunakan merupakan hasil percobaan pada pengamatan proteksi radiasi sel. Obyek yang akan diklasifikasikan adalah senyawa pembentuk suatu obat. Senyawa tersebut diklasifikasikan kedalam kategori positif (proteksi radiasi tinggi) dan negatif (proteksi radiasi rendah). Jumlah data yang dianalisis adalah sebanyak 84 senyawa dan 217 variabel. Hasil analisis metode Lorens dan ELR yang didapatkan selanjutnya dibandingkan dengan menggunakan kriteria total akurasi, *Balanced Classification Rate* (BCR), dan *Area Under Curve* (AUC). Selain itu juga dilakukan analisis simulasi denga tiga tujuan, yaitu untuk mengetahui efek penambahan jumlah variabel dengan rasio $n:p$ adalah sebesar 1:2, 1:3, …, 1:10, efek keseimbangan data dengan rasio kategori positif dan negatifnya adalah 1:1, 1:20, dan 1:100, dan efek multikolinieritas. Hasil analisis simulasi menunjukkan bahwa pada studi simulasi, ELR lebih unggul dalam mengatasi efek penambahan variabel, efek keseimbangan data, dan efek multikolinieritas data. Akan tetapi pada analisis data riil, Lorens memiliki performansi yang lebih baik daripada ELR dengan total akurasi sebesar 69,41%, BCR sebesar 67,64%, dan AUC sebesar 0,7306.



**Kata kunci:** *Drug Discovery*, *Ensemble Logistic Regression* (ELR), *Logistic Regression Ensemble* (LORENS), Proteksi Radiasi Sel

*(Halaman ini sengaja dikosongkan)*

# *CLASSIFICATION OF HIGH DIMENTIONAL DATA USING ENSEMBLE BASED METHOD OF LOGISTIC REGRESSION APPLIED TO DRUG DISCOVERY*


*Name* : T. Dwi Ary Widhianingsih
*Student's Identity Number* : 06211650010024
*Supervisors* : Dr.rer.pol. Heri Kuswanto, S.Si, M.Si
Dr.rer.pol. Dedy Dwi Prastyo, S.Si, M.Si


## *ABSTRACT*


*Binary Logistic Regression is one of the most used classification method. However, its application faces some problems, such as multicolinearity, overfitting, and complexity in computation. Therefore, there are so many developments of that method to resolve these problems. One of them is using ensemble concept, such as Logistic Regression Ensemble (Lorens) and Ensemble Logistic Regression (ELR). Lorens is made using ensemble concept which divides variables of the data to some subspaces randomly, so that these subspaces are mutually exclusive. The final target prediction of Lorens is obtained based on probability mean or majority voting. While ELR is built using iterative ensemble that does model training until reaching the convergence criteria. Because of having a basic method and developed from the same concept, in this thesis did the comparison of Lorens and ELR in case of high dimensional data, peculiarly for drug discovery. The dataset used here is about the experiment of radioprotection of cells. The object that would be classified is the drug-forming compounds. These compounds are classified to either positive (high radioprotection) or negative (low radioprotection) category. The properties of the cells that have been penetrated using the compound are used as the predictors. The number of compounds in the dataset are 84 and the properties are 217. To decide the better method, we calculate total accuracy, Balanced Classification Rate (BCR), and Area Under the Curve (AUC). Beside the real data analysis, we also do simulation to perform the effect of the increasing number of variables with $n$: $p$ ratio is 1:2, 1:3, …, 1:10, the increasing of imbalance ratio using 1:1, 1:20, and 1:100 ratio, and multicolinearity. The result of simulation study shows that ELR outperforms Lorens to resolve these three effects. Otherwise, the real data analysis shows that Lorens has the better performance than ELR with total accuracy 69.41%, BCR 67.64%, and AUC 0.7306.*



***Keywords:*** *Drug Discovery, Ensemble Logistic Regression (ELR), Logistic Regression Ensemble (Lorens), Radioprotection*

*(Halaman ini sengaja dikosongkan)*

# KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Tuhan Yang Maha Esa. Atas segala anugerah-Nya, penulis dapat menyelesaikan penulisan buku Tesis dengan judul "**Klasifikasi Data Berdimensi Tinggi dengan Metode *Ensemble* Berbasis Regresi Logistik dalam Permasalahan *Drug Discovery***" dengan baik dan lancar.

Penulisan Tesis ini adalah salah satu syarat yang harus dipenuhi dalam memperoleh gelar Magister sesuai dengan kurikulum Departemen Statistika, Fakultas Matematika, Komputasi, dan Sains Data, Institut Teknologi Sepuluh Nopember, Surabaya. Dalam penyelesaian Tesis serta laporan ini penulis tidak terlepas dari bantuan serta dukungan dari berbagai pihak. Oleh karena itu penulis ingin mengucapkan terima kasih sebesar-besarnya kepada:

1. Orang tua yang telah memberikan dukungan penuh berupa motivasi dan finansial untuk memenuhi segala kebutuhan yang diperlukan selama mengerjakan Tesis.
2. Dosen pembimbing, yaitu Pak Heri Kuswanto dan Pak Dedy Dwi Prastyo yang senantiasa meluangkan waktu untuk berdiskusi, memberikan saran, dan memberikan segala dukungan untuk kelancaran penyelesaian Tesis.
3. Dosen Penguji, yaitu Bu Santi Wulan Purnami dan Bu Santi Puteri Rahayu yang telah memberikan kritik dan saran untuk membangun Tesis yang dikerjakan agar semakin baik dan berisi.
4. Seluruh Dosen di Departemen Statistika yang telah memberikan ilmu selama masa perkuliahan.
5. Staf administrasi Program Studi Pascasarjana, Mbak Fathin Hilmiyah, yang telah membantu dalam proses pemberkasan yang berkaitan untuk penyelesaian Tesis.
6. Teman-teman S2 Statistika ITS angkatan 2016, khususnya teman seperjuangan: Mbak Tri, Mbak Saidah, Mbak Kiki, Mbak Prizka, Mbak Asri, Meranggi, Alfan, dkk. yang telah menjadi teman yang baik, selalu sabar, dan selalu memberikan motivasi-motivasi selama masa perkuliahan dan selama masa pengerjaan Tesis.

7. Endar Alam, yang telah memberikan motivasi dan dukungan untuk segera menyelesaikan Tesis.
8. Pihak-pihak lain yang telah mendukung dan membantu dalam penyusunan Tesis ini yang tidak dapat disebutkan satu per satu. Terima kasih.

Sekali lagi penulis mengucapkan banyak-banyak terima kasih kepada semua pihak yang telah banyak mendukung dalam penyelesaian Tesis ini. Penulis menyadari bahwa penyusunan Tesis ini masih jauh dari sempurna, maka kritik dan saran yang membangun akan senantiasa penulis harapkan. Semoga laporan ini dapat memberikan sedikit informasi dan ilmu yang bermanfaat bagi semua pihak.

Surabaya, Juli 2018

**Penulis**

# DAFTAR ISI

# DAFTAR TABEL

*(Halaman ini sengaja dikosongkan)*

# DAFTAR GAMBAR

# DAFTAR LAMPIRAN

*(Halaman ini sengaja dikosongkan)*

# BAB 1
# PENDAHULUAN

## 1.1. Latar Belakang

Analisis klasifikasi merupakan metode multivariat yang berhubungan dengan pemisahan observasi dan pengalokasian observasi baru ke dalam kategori data. Tujuan yang harus dicapai dalam analisis klasifikasi adalah mendapatkan fungsi diskriminasi yang dapat memisahkan observasi semaksimal mungkin dan mendapatkan *rule* (aturan) yang dapat digunakan untuk menentukan kategori pada observasi yang baru (Johnson dan Wichern, 2007). Beberapa metode yang dapat digunakan dalam analisis klasifikasi adalah regresi logistik, analisis diskriminan, *Support Vector Machine* (SVM), *Artificial Neural Network* (ANN), *Naïve Bayes Classifier* (NBC), *random forest*, dan lain-lain. Diantara metode-metode tersebut, regresi logistik merupakan salah satu metode yang cukup sering digunakan dalam permasalahan klasifikasi.

Regresi logistik dikembangkan dari rumpun metode parametrik *Generalized Linear Model* (GLMs). Dalam permasalahan analisis klasifikasi data dikotomus, metode ini sering disebut sebagai regresi logistik biner. Regresi logistik biner merupakan metode standar yang cukup banyak digunakan, karena model klasifikasi yang dihasilkan bisa diinterpretasikan. Selain itu, fungsi tujuan pada metode ini dapat menghasilkan nilai probabilitas yang bisa digunakan untuk mengetahui kecenderungan suatu obyek terhadap kategori tertentu, misalnya kategori positif dan negatif. Walaupun demikian, regresi logistik biner dapat memberikan hasil yang kurang baik apabila diterapkan pada data berdimensi tinggi, yang dalam hal ini didefinisikan sebagai data yang banyaknya variabel lebih besar daripada banyaknya observasi ($p \gg n$). Bielza, dkk. (2011) menjelaskan 4 permasalahan yang akan dihadapi jika regresi logistik biner diaplikasikan pada data berdimensi tinggi. Pertama, terdapat solusi yang tidak unik dalam perhitungan estimasi parameter regresi karena proses estimasinya dilakukan dengan menggunakan data yang sangat sedikit (relatif terhadap banyaknya variabel). Kedua, terdapat permasalahan multikolinearitas, yaitu korelasi antarvariabel

prediktornya tinggi. Banyaknya variabel yang semakin bertambah akan memperbesar kemungkinan adanya pola kombinasi linier antarvariabel. Ketiga, terjadi fenomena *overfitting*, artinya model yang dihasilkan memiliki performansi yang sangat baik pada data *training*, tetapi menghasilkan prediksi yang buruk pada data *testing*. *Overfitting* terjadi ketika model yang terbentuk bersifat sangat kompleks. Kompleksitas model terbentuk dari banyaknya variabel prediktor yang melebihi banyaknya data yang digunakan (Romero, dkk., 2011). Keempat, adanya kompleksitas dalam hal komputasi, misalnya dalam perhitungan estimasi parameter. Metode numerik untuk menghitung nilai estimasi ini membutuhkan komputasi tingkat tinggi karena harus menyelesaikan perhitungan yang rumit pada setiap iterasinya, sebagai contoh dalam perhitungan invers matrik pada metode *newton raphson*. Beberapa keterbatasan regresi logistik biner tersebut mengakibatkan metode ini tidak dapat diaplikasikan dengan baik pada data berdimensi tinggi.

Metode untuk penyelesaian permasalahan regresi logistik biner saat ini banyak dikembangkan. Beberapa solusi diantaranya adalah seleksi variabel, ekstraksi variabel, dan regularisasi (Bielza, dkk., 2011). Selain itu, peneliti juga mengembangkan metode *ensemble* untuk mengatasi permasalahan tersebut. Secara umum, konsep *ensemble* telah diterapkan pada berbagai jenis permasalahan dalam statistika, seperti *time series* (Suhartono, dkk., 2012), pemodelan regresi (Shu dan Burn, 2004), dan pemodelan dengan pendekatan bayesian (Duan, dkk., 2007). Dalam analisis klasifikasi, metode *ensemble* dapat diartikan sebagai suatu model atau *rule* yang dibentuk dari sekumpulan model klasifikasi, sehingga dapat memisahkan data kedalam kategori yang berbeda dengan menggunakan nilai kombinasi hasil prediksi dari masing-masing model (Dietterich, 2000). Dalam prosesnya, metode *ensemble* dapat memperbaiki hasil analisis klasifikasi yang dibentuk dari model yang memiliki performansi kurang baik (Rokach, 2010). Selain itu, metode *ensemble* juga dapat memperbaiki nilai estimasi atau *rule* yang tidak stabil (Bühlmann, 2012). Kelebihan-kelebihan tersebut merupakan beberapa alasan dari banyaknya penerapan metode *ensemble* dalam pengembangan dan modifikasi metode-metode klasifikasi, salah satunya adalah regresi logistik biner. Lim pada tahun 2007 mengembangkan *Logistic Regression Ensemble* (Lorens) dan Zakharov

dan Dupont pada tahun 2011 mengembangkan metode *Ensemble Logistic Regression* (ELR). Kedua metode tersebut dibentuk menggunakan algoritma *ensemble* yang berbeda.

Lorens menerapkan konsep *ensemble* yang melakukan partisi variabel menjadi beberapa *subspace* (pembagian variabel dalam beberapa kelompok) secara random dan saling bebas dengan jumlah yang seimbang. Pada tiap-tiap *subspace*, pemodelan dilakukan menggunakan regresi logistik biner, kemudian hasilnya digabungkan untuk menghitung nilai prediksi akhir, baik dengan cara perhitungan rata-rata maupun dengan *majority voting* (penentuan hasil prediksi dengan jumlah *vote* terbanyak). Lim, dkk. (2009) menjelaskan bahwa dengan menggunakan metode Lorens, proses seleksi variabel tidak perlu dilakukan, karena pemodelan telah diaplikasikan pada data yang memiliki dimensi lebih rendah (*subspace*). Lorens telah terapkan pada beberapa kasus, seperti yang dilakukan oleh Kuswanto, dkk. (2015) untuk prediksi perilaku pelanggan dan dihasilkan total akurasi prediksi sebesar 66% hingga 77%, Kuswanto dan Werdhana (2017) untuk analisis klasifikasi ekspresi gen pada penyakit alzheimer dan menghasilkan total akurasi prediksi sebesar 75,28% dan *Area Under Curve* (AUC) sebesar 0,759, serta Kuswanto, dkk. (2018) pada analisis klasifikasi inhibitor enzim dan menghasilkan total akurasi prediksi sebesar 88,95%, 92,1%, dan 100% untuk tiga *dataset* mengenai pengamatan enzim aofb,cah2, dan hs90a.

Konsep *ensemble* pada metode ELR adalah melakukan pemodelan secara iteratif. Dengan demikian, model yang digunakan untuk prediksi dihasilkan dari model akhir yang dibentuk. Metode ini dapat dikategorikan kedalam metode *embedded*, karena proses estimasi parameter dalam algoritmanya dilakukan secara bersamaan dengan proses seleksi variabel (Zakharov dan Dupont, 2011). Metode ELR pernah diaplikasikan dalam permasalahan deteksi kantuk pada pengemudi kendaraan bermotor oleh Kannanthanathu (2017) dan menghasilkan akurasi prediksi sebesar 90% hingga 95%.

Pengembangan regresi logistik menjadi Lorens dan ELR meningkatkan kemampuan pengaplikasian metode tersebut untuk data berdimensi tinggi. Data berdimensi tinggi dapat ditemukan dalam bidang farmasi dan bioinformatika, misalnya dalam kasus *drug discovery*. Pada kasus ini, terdapat proses penemuan

senyawa-senyawa baru yang nantinya akan diseleksi untuk mendapatkan kandidat sebagai komposisi dasar dalam pembuatan suatu obat. Dalam prosesnya, suatu senyawa yang ditemukan akan digunakan sebagai perlakuan yang dikenakan kedalam sel dan efek yang ditimbulkan dari perlakuan ini akan diukur. Pengukuran efek ini dilakukan dengan mengamati karakteristik sel dengan variabel yang sangat beragam, seperti karakteristik yang berkaitan dengan sifat hidrofobisitas sel, struktur sel, dan ukuran sel. Akan tetapi, untuk menemukan senyawa yang sesuai dengan tujuan tertentu dari suatu percobaan, proses yang dilakukan sangat rumit dan membutuhkan waktu yang lama serta biaya yang cukup mahal. Oleh karena itu dalam permasalahan *drug discovery* ini, banyaknya senyawa yang ditemukan sebagai perlakuan tidak sebanyak karakteristik yang akan diukur. Dengan demikian, *dataset* yang dihasilkan dalam *drug discovery* dapat dikategorikan sebagai data berdimensi tinggi. Gmuender (2002) menjelaskan bahwa saat ini terdapat banyak senyawa baru yang telah dikembangkan untuk kepentingan dalam bidang biologi. Banyaknya senyawa yang ditemukan ini mempengaruhi perkembangan penelitian dalam kasus *drug discovery*. Seiring bertambahnya penemuan baru tersebut, permasalahan lain juga semakin banyak bermunculan. Salah satunya adalah dalam proses pemilihan kandidat senyawa sebagai komposisi dasar pembuatan obat baru. Proses ini memakan waktu yang cukup lama jika dilakukan secara manual. Oleh karena itu, para peneliti mengembangkan penerapan metode *machine learning* untuk mengatasi permasalahan tersebut, yaitu dengan tujuan agar proses pemilihan kandidat senyawa dapat dilakukan secara lebih efektif dan efisien. Beberapa metode *machine learning* yang pernah diterapkan untuk permasalahan *drug discovery* adalah SVM (Burbidge, dkk., 2001; Warmuth, dkk., 2003; Alvarsson, dkk., 2016), ANN (Cheng dan Sutariya, 2012; Scotti, dkk., 2015; Huynh, dkk., 2016), *random forest* (Jayaraj, dkk., 2016), *improved naïve bayesian algorithm* (Bai, dkk., 2018), dan *weighted nearest neighbour* (Laarhoven dan Marchiori, 2013).

Pada penelitian ini, dilakukan analisis klasifikasi berdimensi tinggi dengan studi kasus *drug discovery*. Analisis dilakukan terhadap hasil percobaan mengenai pengamatan proteksi radiasi sel. Data yang digunakan merupakan hasil percobaan yang dilakukan oleh Ariyasu, dkk. pada tahun 2014. Data ini pernah dipakai dalam

analisis yang dilakukan oleh Matsumoto, dkk. (2015) untuk analisis klasifikasi menggunakan metode *random forest* dan SVM dan dihasilkan performansi AUC sebesar 0,581 dan 0,411 pada masing-masing metode. Matsumoto, dkk. (2016) melakukan analisis klasifikasi dengan *dataset* ini menggunakan *random forest* dan SVM dengan seleksi variabel berdasarkan tingkat kepentingannya (*importance*) dan menghasilkan performansi AUC optimal sebesar 0,619 untuk metode *random forest* dengan 10 variabel terpenting dan 0,646 untuk metode SVM dengan 5 variabel terpenting. Sedangkan Kimura, dkk. (2017) melakukan analisis klasifikasi proteksi radiasi sel dengan menggunakan metode *K-Nearest Neighbor* (KNN), *Extreme Gradient Boosting* (XGB), SVM, dan *random forest* dan menghasilkan performansi AUC untuk masing-masing metode sebesar 0,757, 0,635, 0,628, dan 0,688. Obyek yang diklasifikasikan dalam data tersebut adalah senyawa yang digunakan sebagai perlakuan dalam percobaan. Senyawa ini diklasifikasikan kedalam kategori positif dan negatif. Kategori positif menunjukkan efek proteksi radiasi yang tinggi, sedangkan kategori negatif menunjukkan efek proteksi radiasi yang rendah. Metode yang digunakan adalah Lorens dan ELR. Kedua metode memiliki beberapa kesamaan, antara lain kedua metode merupakan pengembangan dari metode regresi logistik biner yang dibentuk dengan menggunakan konsep *ensemble* dan keduanya dibuat dengan tujuan yang sama, yaitu agar metode regresi logistik biner dapat diaplikasikan pada data berdimensi tinggi. Oleh karena itu, dalam penelitian ini ingin diketahui metode yang lebih baik diantara metode Lorens dan ELR khususnya dalam studi kasus *drug discovery*.

## 1.2. Perumusan Masalah

Analisis klasifikasi diaplikasikan pada data yang memiliki variabel respon diskrit yang berskala nominal atau ordinal. Dalam permasalahan analisis klasifikasi pada data dikotomus, metode yang sederhana yang umum digunakan adalah regresi logistik. Metode ini memiliki beberapa kelebihan, yaitu hasil analisis yang didapatkan adalah berupa nilai probabilitas, sehingga penentuan kategori prediksi dapat dilakukan dengan mudah. Akan tetapi, penerapan metode regresi logistik biner memiliki kelemahan pada data berdimensi tinggi. Untuk mengatasi hal ini, pengembangan regresi logistik banyak dilakukan, misalnya dengan memanfaatkan

konsep *ensemble*, seperti Lorens dan ELR. Dengan demikian, pada penelitian ini ingin diketahui bagaimana hasil penerapan dan perbandingan hasil analisis dari kedua metode pengembangan tersebut pada data berdimensi tinggi, khususnya dalam permasalahan *drug discovery* dengan studi kasus mengenai data pengamatan proteksi radiasi sel. Selain itu, ingin diketahui pula bagaimana hasil studi simulasi yang dilakukan untuk metode Lorens dan ELR dalam beberapa permasalahan yang mungkin muncul pada data berdimensi tinggi.

## 1.3. Tujuan Penelitian

Berdasarkan rumusan masalah yang disebutkan, didapatkan tujuan penelitian di bawah ini.

1. Mendapatkan hasil studi simulasi dari analisis klasifikasi data berdimensi tinggi dengan metode Lorens dan ELR
2. Melakukan perbandingan hasil klasifikasi yang didapatkan dari penerapan metode Lorens dan ELR pada data riil mengenai kasus *drug discovery*

## 1.4. Manfaat Penelitian

Sesuai dengan tujuan peneilitan, manfaat yang bisa didapatkan dari penelitian ini disebutkan dalam uraian di bawah ini.

1. Kesimpulan dari studi simulasi dapat memberikan informasi mengenai performansi metode Lorens dan ELR kaitannya dengan analisis data berdimensi tinggi
2. Hasil perbandingan metode yang didapatkan bisa digunakan sebagai referensi dalam analisis selanjutnya mengenai analisis data berdimensi tinggi
3. Model klasifikasi yang didapatkan bisa digunakan untuk pengembangan teknologi baru untuk memprediksi senyawa dengan lebih efektif dan efisien

## 1.5. Batasan Masalah

Batasan masalah pada penelitian ini terkait dengan metode *ensemble* yang digunakan. Metode pengembangan regresi logistik yang digunakan adalah Lorens (*Logistic Regression Ensemble*) dan ELR (*Ensemble Logistic Regression*). Kedua metode ini diterapkan pada data yang memiliki sifat berdimensi tinggi, yaitu dalam

studi kasus *drug discovery* mengenai pengamatan proteksi radiasi sel yang percobaannya dilakukan untuk mengatasi efek negatif yang muncul pada sel-sel normal di sekitar sel kanker. Selain itu, studi simulasi yang dilakukan dalam penelitian ini dibatasi hanya untuk mengetahui efek penambahan jumlah variabel, efek rasio *imbalance*, dan efek multikolinieritas. Karakteristik data yang digunakan untuk mambangkitkan data tidak diambil dari data *drug discovery* (atau data riil yang digunakan). Variabel yang dibangkitkan adalah sebanyak dua jenis, yaitu diskrit dan kontinyu yang masing-masing dibangkitkan dengan distribusi binomial dan distribusi normal.

*(Halaman ini sengaja dikosongkan)*

# BAB 2
# TINJAUAN PUSTAKA

Pembahasan dalam bagian tinjauan pustaka pada penelitian ini menguraikan beberapa metode yang mendukung analisis yang dilakukan, seperti regresi logistik, baik regresi logistik biner maupun regresi logistik terregularisasi, metode Lorens dan ELR, serta metode untuk proses evaluasi dan penilaian performansi hasil analisis klasifikasi.

## 2.1 Regresi Logistik

Metode regresi merupakan metode pemodelan yang paling umum digunakan yang berfokus pada identifikasi pola hubungan antara variabel respon dengan variabel prediktor. Jika variabel respon yang terdapat dalam data berupa variabel kategorik, maka metode regresi yang sesuai adalah regresi logistik. Hosmer dan Lemeshow (2000) menyatakan bahwa metode ini merupakan metode standar yang banyak digunakan untuk pemodelan data diskrit. Pada bagian ini dijelaskan mengenai regresi logistik biner dan regresi logistik yang terregularisasi.

### 2.1.1 Regresi Logistik Biner

Metode standar yang cukup banyak digunakan untuk memodelkan data dikotomus adalah regresi logistik biner. Dimisalkan terdapat matrik $\boldsymbol{X}$ yang berukuran $n \times p$ yang berisi variabel prediktor dengan $\boldsymbol{x}_i = \left(x_{i1}, x_{i2}, \dots, x_{ip}\right)^T$ untuk $i = 1,2, \dots, n$ dan vektor $\boldsymbol{y}$ yang berukuran $n \times 1$ yang berisi variabel respon biner dengan $Y$ dikotomus dengan nilai yang mungkin adalah $\{0,1\}$, maka fungsi $\pi(\boldsymbol{x}_i)$ atau $\pi\left(x_{i1}, x_{i2}, \dots, x_{ip}\right)$ untuk $Y_i = 1$ ditunjukkan dalam Persamaan (2.1).

$$\begin{aligned}\pi(\boldsymbol{x}_i) &= P(Y_i = 1|\boldsymbol{x}_i) \\ &= 1 - P(Y_i = 0|\boldsymbol{x}_i)\end{aligned} \tag{2.1}$$

dengan $\pi(\boldsymbol{x}_i)$ disebut sebagai model regresi logistik biner yang memiliki rentang nilai antara 0 sampai 1. Jika kategori 1 digunakan sebagai acuan, maka fungsi $P(Y_i = 1|\boldsymbol{x}_i)$ secara umum dapat dituliskan sebagai berikut.

$$P(Y_i = 1|\boldsymbol{x}_i) = \frac{1}{1 + e^{-(\beta_0 + x_i^T \boldsymbol{\beta})}} \tag{2.2}$$

dengan vektor $\boldsymbol{\beta} = (\beta_1, \beta_2, \dots, \beta_p)^T$ dan $y_i \in \{-1,1\}$ untuk $i = 1,2, \dots, n$. Sedangkan fungsi probabilitas untuk $Y_i = 0$ adalah sebagai berikut.

$$P(Y_i = 0|\boldsymbol{x}_i) = \frac{e^{\beta_0 + x_i^T \boldsymbol{\beta}}}{1 + e^{\beta_0 + x_i^T \boldsymbol{\beta}}} = \frac{1}{1 + e^{\beta_0 + x_i^T \boldsymbol{\beta}}} \tag{2.3}$$

Persamaan (2.2) dapat dibentuk menjadi fungsi logit, yaitu dengan cara melakukan transformasi logaritma natural terhadap rasio $\pi(\boldsymbol{x}_i)$ dengan $1 - \pi(\boldsymbol{x}_i)$, sehingga didapatkan $\ln\left(\frac{\pi(\boldsymbol{x}_i)}{1-\pi(\boldsymbol{x}_i)}\right) = -(\beta_0 + \boldsymbol{x}_i^T \boldsymbol{\beta})$.

Fungsi distribusi pada model regresi logistik biner mengikuti distibusi *bernoulli*. Dengan didefinisikan fungsi $\pi(\boldsymbol{x}_i)$ sebagai peluang kategori 1, maka fungsi densitas yang didapatkan adalah sebagai berikut.

$$f(y_i|\beta_0, \boldsymbol{\beta}, \boldsymbol{x}_i) = \pi(\boldsymbol{x}_i)^{y_i}\left(1 - \pi(\boldsymbol{x}_i)\right)^{1-y_i} \tag{2.4}$$

Berdasarkan Persamaan (2.4), persamaan *maximum likelihood* yang bisa didapatkan adalah sebagai berikut.

$$\mathcal{L}(\beta_0, \boldsymbol{\beta}|\boldsymbol{x}_i, y_i) = \prod_{i=1}^{n} \pi(\boldsymbol{x}_i)^{y_i}\left(1 - \pi(\boldsymbol{x}_i)\right)^{1-y_i} \tag{2.5}$$

$$= \prod_{i=1}^{n} \left(\frac{\pi(\boldsymbol{x}_i}{1 - \pi(\boldsymbol{x}_i)}\right)^{y_i} \left(1 - \pi(\boldsymbol{x}_i)\right) \tag{2.6}$$

Untuk mendapatkan estimasi parameter dengan cara *maximum likelihood*, fungsi Persamaan (2.6) dapat dijabarkan seperti langkah berikut ini.

$$\mathcal{L}(\beta_0, \boldsymbol{\beta}|\boldsymbol{x}_i, y_i) = \prod_{i=1}^{n} \left(\frac{\pi(\boldsymbol{x}_i}{1 - \pi(\boldsymbol{x}_i)}\right)^{y_i} \left(1 - \pi(\boldsymbol{x}_i)\right)$$

$$= \prod_{i=1}^{n} \left(\frac{1}{1 + e^{-(\beta_0 + \boldsymbol{x}^T \boldsymbol{\beta})}} \cdot \frac{1 + e^{-(\beta_0 + \boldsymbol{x}^T \boldsymbol{\beta})}}{e^{-(\beta_0 + \boldsymbol{x}^T \boldsymbol{\beta})}}\right)^{y_i} \left(\frac{e^{-(\beta_0 + \boldsymbol{x}^T \boldsymbol{\beta})}}{1 + e^{-(\beta_0 + \boldsymbol{x}^T \boldsymbol{\beta})}}\right)$$

$$= \prod_{i=1}^{n} \left(\frac{1}{e^{-(\beta_0 + \boldsymbol{x}^T \boldsymbol{\beta})}}\right)^{y_i} \left(\frac{e^{-(\beta_0 + \boldsymbol{x}^T \boldsymbol{\beta})}}{1 + e^{-(\beta_0 + \boldsymbol{x}^T \boldsymbol{\beta})}}\right)$$

$$= \prod_{i=1}^{n} \left(e^{\beta_0 + \boldsymbol{x}^T \boldsymbol{\beta}}\right)^{y_i} \left(\frac{1}{1 + e^{\beta_0 + \boldsymbol{x}^T \boldsymbol{\beta}}}\right)$$

$$\mathcal{L}(\beta_0, \boldsymbol{\beta}|\boldsymbol{x}_i, y_i) = \prod_{i=1}^{n}\left(e^{\beta_0+\boldsymbol{x}^T\boldsymbol{\beta}}\right)^{y_i}\left(1+e^{\beta_0+\boldsymbol{x}^T\boldsymbol{\beta}}\right)^{-1} \tag{2.7}$$

Estimator yang didapatkan dari hasil *maximum likelihood* bersifat tidak linier (Bielza, dkk., 2011), sehingga digunakan metode iteratif untuk menyelesaikannya. Metode yang cukup sering digunakan untuk mendapatkan nilai estimasi parameter model regresi logistik adalah Newton Raphson.

Newton Raphson merupakan salah satu metode yang digunakan untuk secara numerik menyelesaikan persamaan yang memiliki solusi tidak *closed-form*, seperti solusi $\hat{\beta}_0$ dan $\widehat{\boldsymbol{\beta}}$ pada regresi logistik. Proses estimasi nilai $\beta_0$ dan $\boldsymbol{\beta}$ dengan metode ini mengikuti persamaan berikut ini (Lee, dkk., 2006).

$$\left\{\hat{\beta}_0, \widehat{\boldsymbol{\beta}}\right\}^{(k)} = \left\{\hat{\beta}_0, \widehat{\boldsymbol{\beta}}\right\}^{(k-1)} - \boldsymbol{H}^{-1}\left(\beta_0^{(k)}, \boldsymbol{\beta}^{(k)}\right)\boldsymbol{\nabla}\ell\left(\beta_0^{(k)}, \boldsymbol{\beta}^{(k)}\right) \tag{2.8}$$

dengan $k$ menunjukkan indeks iterasi pada proses estimasi dengan Newton Raphson (Li, dkk., 2015). Vektor $\boldsymbol{\nabla}\ell\left(\beta_0^{(k)}, \boldsymbol{\beta}^{(k)}\right)$ menunjukkan vektor gradien yang merupakan turunan pertama dari fungsi ln-*likelihood*, sedangkan matrik $\boldsymbol{H}^{-1}\left(\beta_0^{(k)}, \boldsymbol{\beta}^{(k)}\right)$ adalah matrik hessian untuk $\ell(\beta)$. Berdasarkan fungsi *likelihood* yang dituliskan pada Persamaan (2.5), maka fungsi ln-*likelihood* dapat didapatkan seperti langkah di bawah ini.

$$\begin{aligned}\ell(\beta_0, \boldsymbol{\beta}) &= \ln\left\{\prod_{i=1}^{n}\pi(\boldsymbol{x}_i)^{y_i}\left(1-\pi(\boldsymbol{x}_i)\right)^{1-y_i}\right\} \\ &= \sum_{i=1}^{n} y_i \ln\pi(\boldsymbol{x}_i) + (1-y_i)\ln\left(1-\pi(\boldsymbol{x}_i)\right)\end{aligned} \tag{2.9}$$

Dengan $j^* = 0,1,2,\ldots,p$, maka elemen vektor gradien dan matrik hessian adalah sebagai berikut.

$$\boldsymbol{\nabla}\ell(\beta_0, \boldsymbol{\beta}) = \begin{pmatrix} \frac{\partial l\left(\beta_{j^*}\right)}{\partial \beta_0} \\ \frac{\partial l\left(\beta_{j^*}\right)}{\partial \beta_1} \\ \vdots \\ \frac{\partial l\left(\beta_{j^*}\right)}{\partial \beta_p} \end{pmatrix} \text{ dan } \boldsymbol{H}(\beta_0, \boldsymbol{\beta}) = \begin{pmatrix} \frac{\partial^2 l\left(\beta_{j^*}\right)}{\partial \beta_0^2} & \frac{\partial^2 l\left(\beta_{j^*}\right)}{\partial \beta_0 \beta_1} & \cdots & \frac{\partial^2 l\left(\beta_{j^*}\right)}{\partial \beta_0 \beta_p} \\ \frac{\partial^2 l\left(\beta_{j^*}\right)}{\partial \beta_0 \beta_1} & \frac{\partial^2 l\left(\beta_{j^*}\right)}{\partial \beta_1^2} & \cdots & \frac{\partial^2 l\left(\beta_{j^*}\right)}{\partial \beta_1 \beta_p} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 l\left(\beta_{j^*}\right)}{\partial \beta_0 \beta_p} & \frac{\partial^2 l\left(\beta_{j^*}\right)}{\partial \beta_1 \beta_p} & \cdots & \frac{\partial^2 l\left(\beta_{j^*}\right)}{\partial \beta_p^2} \end{pmatrix}$$

Nilai varians $\boldsymbol{\beta}^* = (\beta_0, \beta_1, \dots, \beta_p)$ didapatkan dari turunan kedua fungsi ln-*likelihood*. Dimisalkan $\frac{\partial^2 l(\boldsymbol{\beta}^*)}{\partial \beta_{j^*}^2}$ merupakan elemen diagonal dari matrik $\boldsymbol{I}(\boldsymbol{\beta}^*)$ dan $\frac{\partial^2 l(\boldsymbol{\beta}^*)}{\partial \beta_{j^*} \partial \beta_{j'}}$ merupakan elemen selain diagonal dari matrik $\boldsymbol{I}(\boldsymbol{\beta}^*)$ dengan $j^*, j' = 0,1,2, \dots, p$ dan $j^* \neq j'$, maka $Var(\boldsymbol{\beta}^*) = \boldsymbol{I}^{-1}(\boldsymbol{\beta}^*)$. Nilai estimasi dari varians $\beta_{j^*}$ dinotasikan dengan $\widehat{Var}(\widehat{\boldsymbol{\beta}^*}) = \hat{\boldsymbol{I}}^{-1}(\widehat{\boldsymbol{\beta}^*}) = \boldsymbol{X}^T \boldsymbol{V} \boldsymbol{X}$ (Hosmer dan Lemeshow, 2000). Elemen matrik $\boldsymbol{X}$ didefinisikan sebagai berikut.

$$\boldsymbol{X} = \begin{pmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1p} \\ 1 & x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix}$$

Sedangkan elemen matrik $\boldsymbol{V}$ adalah sebagai berikut.

$$\boldsymbol{V} = \begin{pmatrix} \hat{\pi}_1(1-\hat{\pi}_1) & 0 & \dots & 0 \\ 0 & \hat{\pi}_1(1-\hat{\pi}_1) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \hat{\pi}_1(1-\hat{\pi}_1) \end{pmatrix}$$

Penentuan signfikansi estimator $\widehat{\boldsymbol{\beta}^*}$ didapatkan dari pengujian Wald dengan rumusan statistik uji sebagai berikut.

$$W_{j^*} = \frac{\hat{\beta}_{j^*}}{\widehat{SE}(\hat{\beta}_{j^*})} \tag{2.10}$$

dengan $W_{j^*}$ menunjukkan statistik uji Wald, $\hat{\beta}_{j^*}$ merupakan estimasi parameter $\beta_{j^*}$ untuk $j^* = 0,1,2, \dots, p$, dan $\widehat{SE}(\hat{\beta}_{j^*}) = \left(\widehat{Var}(\hat{\beta}_{j^*})\right)^{\frac{1}{2}}$ merupakan *standard error* estimator $\hat{\beta}_{j^*}$. Hipotesis uji Wald adalah sebagai berikut.

$H_0 : \beta_{j^*} = 0$ atau variabel ke-$j^* - 1$ tidak signifikan mempengaruhi variabel $y$

$H_1 : \beta_{j^*} \neq 0$ atau variabel ke-$j^* - 1$ signifikan mempengaruhi variabel $y$

Daerah penolakan uji Wald adalah jika $P\left(z_{\frac{\alpha}{2}} \leq \left|W_{j^*}\right|\right)$ kurang dari $\frac{\alpha}{2}$, maka hipotesis nol ditolak, dengan kata lain bahwa variabel ke-$j^*$ dapat dianggap signifikan memengaruhi variabel respon ($y$) secara statistik dengan tingkat signifikansi $\frac{\alpha}{2}$ (Hauck dan Donner, 1977).

Regresi logistik biner saat ini menjadi metode standar yang masih sering digunakan dalam pemodelan analisis klasifikasi karena memiliki beberapa

kelebihan. Kannanthanathu (2017) menjelaskan kelebihan dalam hal implementasi, yaitu memiliki algoritma yang sederhana, sehingga bisa mendapatkan hasil yang lebih cepat dibandingkan dengan metode klasifikasi yang lainnya. Selain itu, regresi logistik biner mampu memberikan akurasi yang cukup baik. Zakharov dan Dupont (2011) menjelaskan bahwa probabilitas yang didapatkan dari hasil pemodelan regresi logistik biner dapat digunakan untuk menentukan kategori atau kelas pada data yang baru atau data *testing*. Oleh karena itu, metode regresi logistik biner digunakan sebagai basis algoritma pemodelan klasifikasi dalam analisis ini.

### 2.1.2 Regresi Logistik Terregularisasi

Permasalahan yang sering terjadi pada regresi logistik biner dalam data berdimensi tinggi adalah terjadinya *overfitting*, yang disebabkan oleh bersarnya variasi estimasi parameter $\beta_0$ dan $\boldsymbol{\beta}$. Oleh karena itu, perlu dilakukan upaya untuk mengatasi permasalahan semacam ini, salah satunya yaitu dengan metode regularisasi. Ng (2004) menyebutkan bahwa terdapat dua jenis regularisasi yang dapat diterapkan untuk metode regresi logistik biner, yaitu regularisasi $l_1$ dan $l_2$. Perbedaan antara kedua jenis regularisasi tersebut adalah bahwa regularisasi $l_1$ menggunakan pinalti dari nilai jumlahan absolut parameter, sedangkan regularisasi $l_2$ menggunakan pinalti dari nilai jumlahan kuadrat parameter. Secara matematis, kedua pinalti ditunjukkan dalam Persamaan (2.11) dan (2.12).

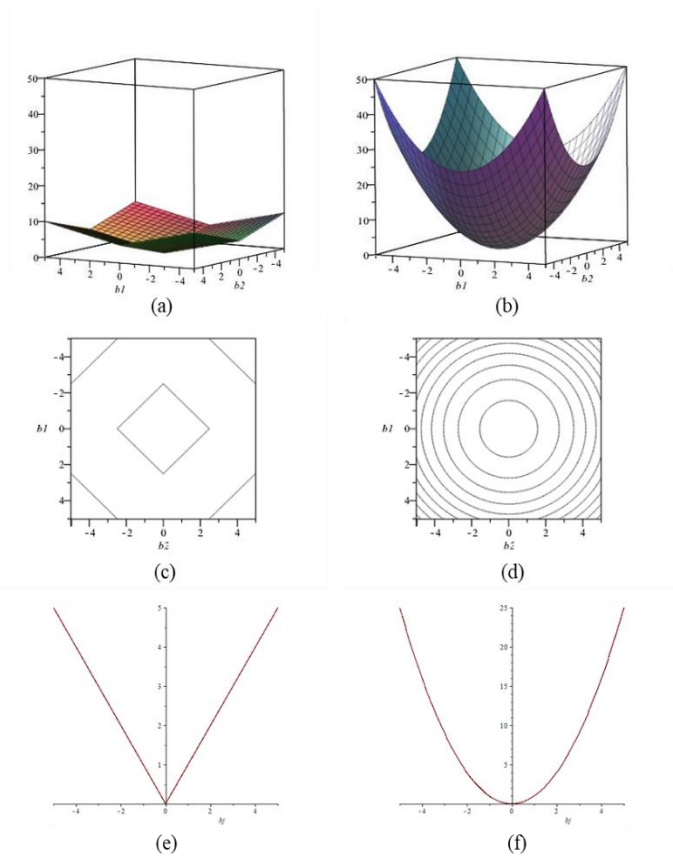
$$R_1(\beta_0, \boldsymbol{\beta}) = \lambda \|\{\beta_0, \boldsymbol{\beta}\}\|_1 = \lambda \left( \beta_0 + \sum_{j=1}^{p} |\beta_j| \right) \tag{2.11}$$

$$R_2(\beta_0, \boldsymbol{\beta}) = \lambda \|\{\beta_0, \boldsymbol{\beta}\}\|_2 = \lambda \left( \beta_0 + \sum_{j=1}^{p} \beta_j^2 \right) \tag{2.12}$$

dengan vektor $\boldsymbol{\beta}$ berukuran $p \times 1$ dan memuat parameter $(\beta_1, \beta_2, \dots, \beta_p)^T$ dan $\lambda$ merupakan parameter regularisasi, dengan $\lambda > 0$. Secara visual, kedua regularisasi ini dapat digambarkan seperti pada Gambar 2.1.

Metode yang menggunakan regularisasi $l_1$ dapat melakukan proses estimasi parameter bersamaan dengan seleksi variabel. Hal ini dikarenakan pada proses estimasi, jika unsur regularisasi pada Persamaan (2.11) diturunkan secara teoritis, maka akan menghasilkan angka nol. Proses ini akan mengakibatkan

estimasi parameter untuk variabel yang memiliki peran kecil dalam pemodelan akan menjadi semakin tidak berarti. Penerapan regularisasi $l_2$ dikembangkan dalam metode LASSO (*Least Absolute Shrinkage and Selection Operator*) (Tibshirani, 1996; Härdle dan Prastyo, 2014). Sedangkan pada regularisasi $l_2$, turunan pertama dari Persamaan (2.12) masih memiliki nilai yang tidak nol, sehingga kegunaannya tidak dapat disamakan dengan regularisasi $l_1$, dengan kata lain metode yang menggunakan regularisasi ini tidak dapat melakukan proses seleksi variabel secara bersamaan dengan proses estimasi parameter model. Pada penerapannya, regulasisasi $l_2$ digunakan untuk mengatasi permasalahan seperti yang diatasi oleh regresi *ridge*, yaitu permasalahan varians estimator yang besar yang dapat menyebabkan estimator tersebut menjadi tidak stabil. Estimator semacam ini dapat menimbulkan perubahan yang signifikan jika pemodelan dilakukan dengan menggunakan data yang berbeda (Montgomery, dkk., 2012).




**Gambar 2.1** Plot Fungsi Regularisasi dengan Domain -5 Sampai 5: (a) dan (b) Regularisasi $l_1$ dan $l_2$ dalam 3 Dimensi untuk Dua Variabel $\beta_1$ dan $\beta_2$, (c) dan (d) Proyeksi Plot Fungsi Regularisasi $l_1$ dan $l_2$ pada bidang 2 Dimensi, sedangkan (e) dan (f) Plot Fungsi Regularisasi $l_1$ dan $l_2$ dalam 2 Dimensi untuk Parameter $\beta_j$

Unsur regularisasi atau pinalti yang ditunjukkan dalam Persamaan (2.11) dan (2.12) ditambahkan pada fungsi *loss* dari model yang digunakan. Dengan demikian, fungsi tujuan metode yang terregularisasi dapat dituliskan seperti pada persamaan berikut ini.

$$f(t) = L(t) + R(t) \tag{2.13}$$

dengan $t$ menunjukkan parameter model, notasi $f(t)$ merupakan fungsi tujuan dari metode terregularisasi, $L(t)$ menunjukkan fungsi *loss* dari metode basis, sedangkan $R(t)$ menunjukkan pinalti atau unsur regularisasi (Andrew dan Gao, 2007).

Fungsi *loss* pada metode regresi logistik biner sama dengan negatif dari fungsi ln-*likelihood* yang ditunjukkan dalam Persamaan (2.9). Fungsi *loss* $L(\beta_0, \boldsymbol{\beta})$ untuk regresi logistik ditunjukkan dalam Persamaan (2.14).

$$L(\pi(\boldsymbol{x}), \hat{\pi}(\boldsymbol{x})) = -\sum_{i=1}^{n} \{y_i \ln \pi(x_i) + (1 - y_i) \ln(1 - \pi(x_i))\} \tag{2.14}$$

dengan membagi fungsi *loss* pada Persamaan (2.14) dengan banyaknya data ($n$), maka didapatkan persamaan untuk nilai rata-rata fungsi *loss* atau *average loss function*, seperti yang ditunjukkan dalam Persamaan (2.15).

$$L_{avg}(\beta_0, \boldsymbol{\beta}) = -\frac{1}{n}\sum_{i=1}^{n} \{y_i \ln \pi(\boldsymbol{x}_i) + (1 - y_i) \ln(1 - \pi(\boldsymbol{x}_i))\} \tag{2.15}$$

Fungsi ln-*likelihood* regresi logistik yang ditunjukkan pada Persamaan (2.9) merupakan fungsi konkaf. Oleh karena berkebalikan dengan fungsi ln-*likelihood*, fungsi $L_{avg}(\beta_0, \boldsymbol{\beta})$ merupakan fungsi konveks (Koh, dkk., 2007). Dengan demikian, proses estimasi parameter pada metode regresi logistik yang terregularisasi dilakukan dengan meminimalkan *average loss function* yang sudah terboboti oleh pinalti $R(\boldsymbol{\beta})$, seperti yang ditunjukkan dalam Persamaan (2.16).

$$\min_{\beta_0, \boldsymbol{\beta}} \{L_{avg}(\beta_0, \boldsymbol{\beta}) + R(\beta_0, \boldsymbol{\beta})\} = \min_{\beta_0, \boldsymbol{\beta}} \left\{ -\sum_{i=1}^{n} \{y_i \ln \pi(\boldsymbol{x}_i) + (1 - y_i) \ln(1 - \pi(\boldsymbol{x}_i))\} + R(\beta_0, \boldsymbol{\beta}) \right\} \tag{2.16}$$

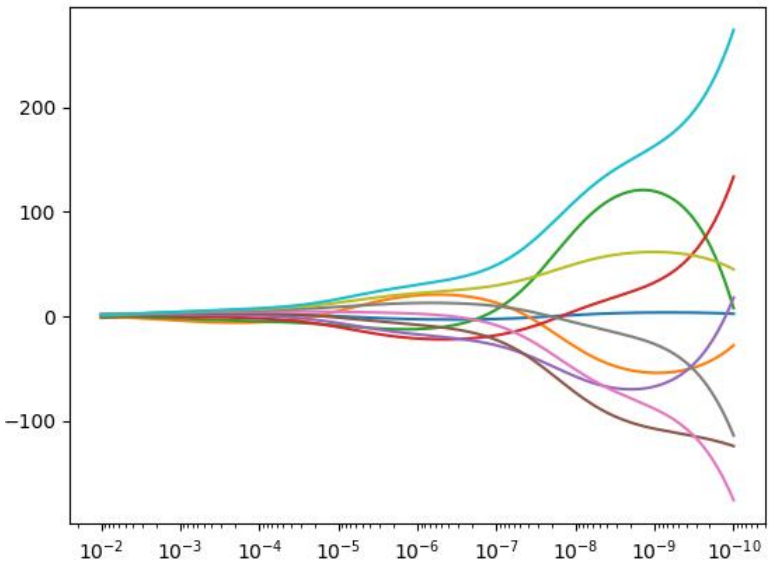
Unsur $R(\boldsymbol{\beta})$ dapat disubtitusi oleh pinalti regularisasi $L_1$ dan $L_2$, yang ditunjukkan dalam Persamaan (2.11) dan (2.12).

Jenis regularisasi yang diterapkan dalam penelitian ini adalah regularisasi $l_2$. Unsur $R(\boldsymbol{\beta})$ dalam regularisasi $l_2$ selanjutnya akan ditambahkan kedalam

rumusan regresi logistik biner. Dengan menggunakan regularisasi $l_2$, maka Persamaan (2.16) dapat ditulis seperti pada Persamaan (2.17).

$$\min_{\beta_0,\boldsymbol{\beta}}\{L_{avg}(\beta_0,\boldsymbol{\beta}) + R_2(\beta_0,\boldsymbol{\beta})\} = \min_{\beta_0,\boldsymbol{\beta}}\left\{-\sum_{i=1}^{n}\{y_i \ln \pi(\boldsymbol{x}_i) + (1-y_i)\ln(1-\pi(\boldsymbol{x}_i))\} + \lambda\left(\beta_0 + \sum_{j=1}^{p}\beta_j^2\right)\right\} \quad (2.17)$$

Efek yang diberikan oleh regularisasi $l_2$ adalah bahwa *standar error* dari estimasi parameter yang dihitung akan semakin mengecil. Akibatnya, perubahan nilai pada estimasi parameter tersebut akan lebih stabil dibandingkan dengan estimasi regresi logistik tanpa regularisasi. Visualisasi efek besarnya angka pada paremeter regularisasi terhadap nilai estimasi parameter regresi logistik ditunjukkan dalam Gambar 2.3. Pada gambar tersebut ditunjukkan bahwa jika nilai $\lambda$ semakin besar (ditampilkan dengan nilai log-lambda), maka nilai estimasi parameter yang didapatkan akan semakin kecil. Untuk menggunakan metode terregularisasi, sebaiknya digunakan nilai $\lambda$ yang optimum. Parameter $\lambda$ merupakan pembobot bagi unsur regularisasi yang ditambahkan dalam *average loss function*, sehingga apabila nilai $\lambda$ yang digunakan terlalu besar, akan mengakibatkan estimasi parameter didominasi oleh unsur regularisasi, akibatnya model yang dihasilkan dapat memiliki kecenderungan underfitting. Dan sebaiknya, jika nilai $\lambda$ semakin mendekati 0, efek regularisasi yang dihasilkan akan semakin kecil, akibatnya *overfitting* dari model yang digunakan tidak dapat diatasi.



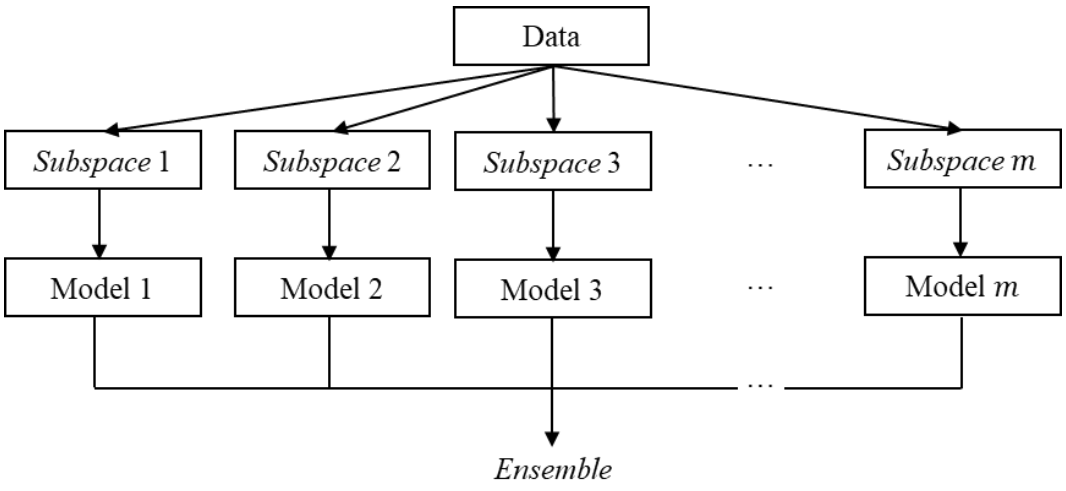

**Gambar 2.2** Visualisasi Regularisasi terhadap Besarnya Nilai Estimasi Parameter Regresi Logistik dengan Sumbu $x$ Menunjukkan Besarnya Nilai $\log(\lambda)$ dan Sumbu $y$ Menunjukkan Besarnya Nilai Estimasi Parameter

## 2.2 Metode *Ensemble*

Metode *ensemble* merupakan salah satu metode yang sedang banyak dikembangkan untuk menyelesaikan permasalahan klasifikasi. Konsep sederhana metode *ensemble* adalah menggabungkan hasil analisis yang dibentuk dari beberapa model klasifikasi. Beberapa pengembangan metode *ensemble* telah dilakukan untuk memperbaiki kelemahan yang terdapat dalam metode klasifikasi yang klasik, seperti regresi logistik biner. Dalam bagian ini, akan dibahas mengenai dua metode hasil pengembangan metode regresi logistik biner dengan menggunakan algoritma *ensemble*, yaitu Lorens dan ELR.

### 2.2.1 *Logistic Regression Ensemble* (Lorens)

Lorens merupakan pengembangan dari metode regresi logistik biner. Metode ini diperkenalkan oleh Lim pada tahun 2007. Ide awal pengembangan Lorens adalah berdasarkan permasalahan bahwa pada metode klasik regresi logistik biner, jika banyaknya variabel lebih besar daripada banyaknya observasi, maka perlu dilakukan seleksi variabel. Dalam penerapan Lorens, proses seleksi variabel dapat diabaikan, karena dalam algoritmanya dilakukan analisis klasifikasi secara parsial untuk setiap *subspace* yang terbentuk, yang pada akhirnya akan digabungkan untuk mendapatkan keputusan secara global. Karena algoritmanya sudah mengakomodasi peran seluruh variabel, maka dalam Lorens tidak perlu dilakukan seleksi variabel (Lim, dkk., 2009).




**Gambar 2.3** Skema Pendekatan CERP yang Diaplikasikan pada Algoritma Lorens

Lorens dikembangkan dengan menggunakan algoritma CERP (*Classification by Ensembles from Random Partitions*), yang sebelumnya telah diperkenalkan oleh Ahn, dkk., (2007). Konsep dasar CERP adalah mengombinasikan sekelompok model klasifikasi yang "lemah" untuk mendapatkan model klasifikasi akurasi prediksi yang lebih baik. Karena melakukan partisi variabel secara random, metode ini dapat mengurangi adanya korelasi antara model klasifikasi (Lim, dkk., 2009). Gambaran algoritma CERP yang diterapkan pada metode Lorens ditunjukkan dalam Gambar 2.3.

Dalam algoritma Lorens, dimisalkan $\boldsymbol{\Theta}$ adalah *space* dari variabel prediktor. Untuk meminimumkan korelasi yang terdapat dalam data, *space* $\boldsymbol{\Theta}$ dipartisi menjadi *M subspace* $(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \dots, \boldsymbol{\theta}_M)$ yang saling bebas dengan jumlah variabel yang seimbang. Model klasifikasi regresi logistik biner dibuat untuk tiap-tiap *subspace* tersebut. Selanjutnya, Lorens menggabungkan hasil dari model klasifikasi yang telah didapatkan untuk setiap *subspace*.

Pada dasarnya, terdapat dua metode penggabungan hasil analisis dari model klasifikasi, yaitu *majority voting* dan perhitungan rata-rata. Penelitian yang telah dilakukan oleh Lim, dkk. (2009) menunjukkan bahwa penggabungan dengan perhitungan rata-rata menghasilkan prediksi yang lebih baik dibandingkan *majority voting*, sehingga *ensemble* dalam analisis ini akan dibuat dengan perhitungan rata-rata hasil analisis klasifikasinya, yaitu rata-rata dari nilai probabilitas. Selanjutnya, rata-rata yang telah dihitung digunakan untuk mendapatkan kelas prediksi, yang ditentukan berdasarkan nilai *threshold* tertentu. Nilai *threshold* yang umum digunakan dalam regresi logistik biner adalah 0,5. Permasalahannya adalah bahwa akurasi hasil prediksi tidak akan reliabel jika proporsi kedua kategori 0 dan 1 tidak seimbang. Oleh karena itu, perhitungan nilai *threshold* pada analisis ini ditentukan berdasarkan rumusan pada Persamaan (2.18).

$$threshold = \frac{\bar{y}+0{,}5}{2} \tag{2.18}$$

dengan $\bar{y}$ adalah rata-rata nilai pada kategori 1 (kategori positif/sukses).

Untuk mendapatkan akurasi yang lebih baik, proses partisi dari *space* $\boldsymbol{\Theta}$ dilakukan secara berulang, sehingga *ensemble* yang terbentuk adalah lebih dari satu. Proses ini dapat menghasilkan multi-*ensemble*. Prediksi dari hasil anggota

multi-*ensemble* ini akan digabungkan dengan cara *majority voting*. Lim, dkk. (2009) menyatakan bahwa perubahan hasil prediksi dapat diabaikan jika banyaknya *ensemble* yang dibentuk lebih dari 10, atau dengan kata lain, 10 *ensemble* sudah dianggap cukup digunakan untuk menentukan prediksi akhir Lorens. Dalam analisis ini untuk menghindari adanya kemungkinan jumlah *vote* yang sama, pengulangan pembuatan partisi dibatasi sampai terbentuk 11 *ensemble*. Sesuai dengan penjelasan di atas, secara singkat algoritma metode Lorens dituliskan dalam Tabel 2.1.

**Tabel 2.1** Algoritma Metode Lorens

| |
|---|
| *Input* 1: Data sampel dengan variabel prediktor $\boldsymbol{X} \in \mathbb{R}^p$ dan variabel respon $\boldsymbol{y} \in \{0,1\}^n$<br>*Input* 2: Jumlah *subspace* $M$, nilai *threshold*, dan jumlah pengulangan $Q$ kali<br>*Output*: Kategori prediksi untuk setiap observasi |
| Algoritma:<br>1. Membuat partisi data kedalam $m$-*subspace* secara random<br>*Do* $m = 1$: $M$<br>2. Memodelkan data tiap-tiap *subspace* dengan regresi logistik biner<br>3. Mendapatkan nilai prediksi probabilitas untuk setiap model pada data *testing*<br>4. Menghitung nilai rata-rata probabilitas untuk *ensemble* yang terbentuk<br>5. Menentukan prediksi kategori dari probabilitas yang didapatkan pada langkah 4 berdasarkan nilai *threshold* yang telah ditentukan<br>*End do*<br>6. Melakukan langkah 1 sampai 5 sampai sebanyak $Q$ kali<br>7. Mendapatkan prediksi kategori dengan *majority voting*. Kategori yang mendapatkan *vote* paling banyak disimpulkan sebagai kategori prediksi |

### 2.2.2 *Ensemble Logistic Regression* (ELR)

Salah satu pengembangan metode regresi logistik biner untuk analisis klasifikasi data berdimensi tinggi adalah metode *Ensemble Logistic Regression* (ELR). Metode ini diperkenalkan oleh Zakharov dan Dupont pada tahun 2011. Algoritma yang dikembangkan dalam metode ini memiliki kemampuan seleksi variabel yang secara bersamaan dilakukan dengan pembuatan model klasifikasi, sehingga dapat dinamakan sebagai metode *embedded*. ELR dibuat dengan menggunakan konsep *ensemble* yang berbasis metode regresi logistik biner terregularisasi.

Metode ELR dibuat dengan berdasarkan regularisasi $l_2$. Regularisasi $l_2$ tidak menjamin untuk mendapatkan model yang *sparse* dan stabil. Untuk mendapatkan model yang seperti ini, proses pemodelan dilakukan dengan menggunakan *subset* variabel, yaitu sebanyak $n$ dari $p$ variabel. Variabel yang digunakan dalam pemodelan dipilih berdasarkan nilai probabilitas yang telah dihitung untuk setiap variabel. Nilai inisial probabilitas didapatkan dari metode $t$-*test ranking* seperti yang ditunjukkan pada Persamaan (2.19).

$$t_j = \frac{\mu_{j+} - \mu_{j-}}{\sqrt{\frac{\sigma_{j+}^2}{n_+} + \frac{\sigma_{j-}^2}{n_-}}} \tag{2.19}$$

dengan $t_j$ adalah nilai statistik uji $t$-*test ranking* untuk variabel ke-$j$ untuk $j = 1,2,\dots,p$. Notasi $\mu_j$ dan $\sigma_j^2$ menunjukkan nilai rata-rata dan varians variabel ke-$j$, dan $n$ menunjukkan banyaknya sampel, sedangkan indeks $+$ dan $-$ menunjukkan kategori 1 dan $-1$. Inisial vektor probabilitas didapatkan berdasarkan hasil perhitungan $1 - p$-*value* dari statistik uji pada Persamaan (2.19). Nilai $p$-*value* ini bisa dihitung atau didapatkan berdasarkan tabel distribusi $t$ dengan derajat bebas sebagai berikut.

$$df = \frac{\sqrt{\frac{\hat{\sigma}_{1+}^2}{n_+} + \frac{\hat{\sigma}_{1-}^2}{n_-}}}{\frac{\left(\frac{\hat{\sigma}_{1+}^2}{n_+}\right)^2}{n_+ - 1} + \frac{\left(\frac{\hat{\sigma}_{1-}^2}{n_-}\right)^2}{n_- - 1}} \tag{2.20}$$

Hasil dari perhitungan $1 - p$-*value* hanya digunakan untuk menentukan nilai awal probabilitas variabel, sedangkan nilai probabilitas pada iterasi berikutnya diperbarui sesuai dengan Persamaan (2.21).

$$prob_{j,l} = \frac{1}{z}\left(prob_{j,l-1} + quality.\beta_j^{2.sign(quality)}\right) \tag{2.21}$$

dengan $l = 1,2,\dots$ menunjukkan iterasi yang dilakukan dan nilai inisial probabilitas $prob_{j,0}$ didapatkan berdasarkan nilai $t_j$. Notasi $z$ menunjukkan *normalized constant* dan $quality$ didefinisikan sebagai nilai kualitas relatif yang dapat dihitung sesuai dengan Persamaan (2.22).

$$quality = \log(1 + BCR_l - \overline{BCR}_{l-1}) \tag{2.22}$$

**Tabel 2.2** *Confusion Matrix*

| | | Prediksi | |
|---|---|---|---|
| | | $P_P$ | $N_P$ |
| Aktual | $P_A$ | *True Positive* ($TP$) | *False Negative* ($FN$) |
| | $N_A$ | *False Positive* ($FP$) | *True Negative* ($TN$) |

Tanda untuk nilai kualitas relatif didapatkan berdasarkan perbandingan nilai $quality$ terhadap nilai $BCR_{l-1}$, yaitu jika $quality > BCR_{l-1}$, maka $sign(quality)$ adalah positif atau $+1$, sedangkan jika $quality < BCR_{l-1}$, maka $sign(quality)$ adalah negatif atau $-1$. BCR atau *Balanced Classification Rate* dihitung berdasarkan Persamaan (2.23).

$$BCR = \frac{1}{2}\left(\frac{TP}{P_A} + \frac{TN}{N_A}\right) \tag{2.23}$$

dengan $TP, TN, P_A,$ dan $N_A$ didapatkan dari *confusion matrix* yang ditunjukkan seperti pada Tabel 2.2.

**Tabel 2.3** Algoritma Metode ELR

| |
|---|
| *Input* 1: Data sampel dengan variabel prediktor $\boldsymbol{X} \in \mathbb{R}^p$ dan variabel respon $\boldsymbol{y} \in \{-1,1\}^n$<br>*Input* 2: Parameter regularisasi $\lambda$ untuk pemodelan regresi logistik biner terregularisasi $L_2$<br>*Input* 3: Inisialisasi vektor $\boldsymbol{prob}_0$ berdasarkan perhitungan *t-test ranking* berdasarkan Persamaan (2.19) dan (2.20)<br>*Input* 4: Inisisalisasi nilai $\overline{BCR}_0 \in [0,1]$<br>*Output*: Model regresi logistik terregularisasi dengan nilai $\overline{BCR}$ yang konvergen |
| Algoritma:<br>*Repeat*<br>1. Membuat partisi data sampel menjadi data *training* dan data *testing* secara *stratified*, sehingga didapatkan $n^*$ data *training*<br>2. Mendapatkan $n^*$ variabel dari $p$ variabel secara random berdasarkan vektor $\boldsymbol{prob}_{l-1}$.<br>3. Membuat model regresi logistik biner terregularisasi $L_2$ untuk data *training* dengan menggunakan $n^*$ variabel terpilih<br>4. Mendapatkan prediksi pada data *testing* dengan menggunakan model dari Langkah 4<br>5. Menghitung nilai $BCR_l$ berdasarkan Persamaan (2.23)<br>6. Menghitung nilai $quality$ berdasarkan Persamaan (2.22)<br>7. Memperbarui vektor $\boldsymbol{prob}_l$ berdasarkan Persamaan (2.21)<br>8. Memperbarui nilai $\overline{BCR}_l = \frac{1}{l+1}(l.\overline{BCR}_{l-1} + BCR_l)$<br>*Until* $\varepsilon = \overline{BCR}_{l-1} - \overline{BCR}_l$ kurang dari nilai konvergensi yang ditentukan |

Perhitungan nilai probabilitas variabel hingga proses mendapatkan nilai $\overline{BCR}_l$ dilakukan secara berulang hingga didapatkan $\overline{BCR}$ yang konvergen. Kriteria konvergensi $\overline{BCR}$ didefinisikan sebagai selisih $\overline{BCR}_l - \overline{BCR}_{l-1}$ atau $\varepsilon$ yang sangat kecil. Algoritma metode ELR secara singkat dituliskan secara singkat dalam Tabel 2.3.

## 2.3 Evaluasi Hasil Analisis Klasifikasi

Penilaian kebaikan model klasifikasi dilakukan dengan menghitung nilai performansinya, misalnya dengan kriteria APER (*Apparent Error Rate*), BCR (*Balanced Classification Rate*), dan AUC (*Area Under Curve*). Salah satu penilaian kebaikan model klasifikasi yang sering digunakan adalah kriteria total akurasi prediksi yang dihitung dari 1-APER. Evaluasi model klasifikasi dengan APER dapat dihitung berdasarkan Persamaan (2.24) disesuaikan dengan *confusion matrix* pada Tabel (2.1).

$$APER = \frac{TP+TN}{TP+FP+TN+FN} \tag{2.24}$$

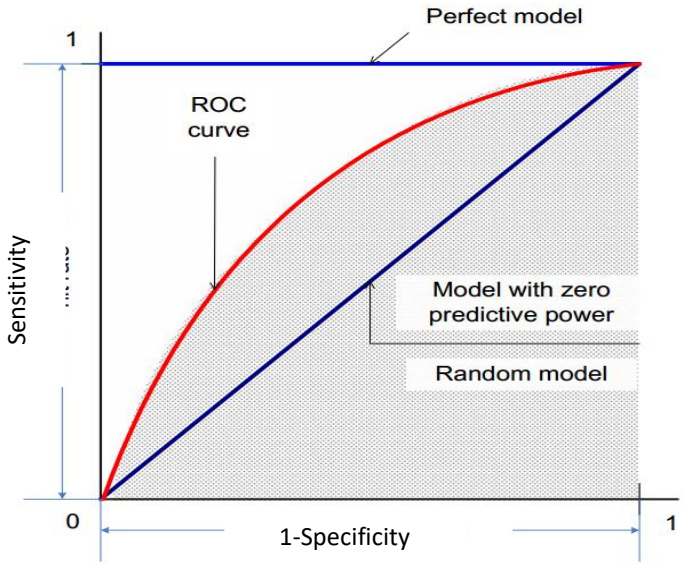
Interpretasi penilaian APER dinyatakan dengan semakin kecil nilai APER, maka model klasifikasi yang didapatkan semakin baik, dan sebaliknya pada interpretasi penilaian total akurasi prediksi. Jika nilai APER mendekati 0 atau total akurasi prediksi mendekati 1 atau 100% pada perhitungan persentasenya, maka model klasifikasi yang didapatkan semakin baik.

Evaluasi dengan menggunakan APER ataupun total akurasi prediksi memiliki kelemahan pada data yang bersifat tidak seimbang (*imbalanced data*) atau data yang mengandung proporsi yang jauh berbeda untuk kedua kategori pada permasalahan data dengan respon dikotomus. Hal ini dikarenakan nilai total akurasi dihitung dari data yang terkategorikan secara benar dibandingkan dengan banyaknya data secara keseluruhan, sehingga tidak dapat mendeteksi bagaimana hasil klasifikasi pada masing-masing kategori secara terpisah. Oleh karena itu digunakan kriteria *sensitivity* dan *specificity* untuk mengetahui akurasi pada masing-masing kategori. Kedua kriteria ini dapat dihitung sesuai dengan Persamaan (2.25) dan (2.26).

$$sensitivity = \frac{TP}{TP+FN} = \frac{TP}{P_A} \tag{2.25}$$

$$specificity = \frac{TN}{TN+FP} = \frac{TN}{N_A} \tag{2.26}$$

Untuk mendapatkan nilai performansi gabungan dari kedua kriteria *sensitivity* dan *specificity* secara bersamaan, perhitungan dapat dilakukan dengan menggunakan BCR, seperti yang telah didefinisikan pada Persamaan (2.23). Interpretasi penilaian BCR dinyatakan bahwa semakin besar nilai BCR, maka model klasifikasi yang didapatkan semakin baik. Jika nilai BCR mendekati 1 atau persentasenya mendekati angka 100%, maka dapat diartikan bahwa hasil klasifikasinya mendekati sempurna.




**Gambar 2.4** Kurva ROC (Härdle, dkk., 2013)

Selain BCR, AUC juga dapat digunakan sebagai perhitungan dari kombinasi nilai *sensitivity* dan *specificity*. Nilai AUC dihitung dari luasan kurva ROC (*Receiver Operating Characteristic*), seperti yang ditunjukkan dalam Gambar 2.5. Kurva ROC dibentuk dari plot antara nilai *sensitivity* dengan $1 - specificity$ yang dihitung dari prediksi kategori berdasarkan semua kemungkinan nilai *threshold* ($\tau$). Luasan di bawah kurva ROC dapat diinterpretasikan sebagai rata-rata kebaikan model untuk setiap nilai $\tau$. Dengan demikian, semakin luas area di bawah kurva ROC, maka semakin baik model klasifikasi yang didapatkan. Nilai AUC maksimum adalah 1, yaitu untuk menunjukkan bahwa model klasifikasi dapat memprediksi data dengan sempurna. Sedangkan nilai AUC 0,5 menunjukkan

bahwa model klasifikasi merupakan model random yang tidak memiliki kemampuan untuk memisahkan data (Härdle, dkk., 2013).

## 2.4 Rasio *Imbalance Data*

Salah satu permasalahan yang banyak muncul pada data berdimensi tinggi adalah adanya kecenderungan data untuk terkategori ke satu kelas tertentu. Hal tersebut yang menyebabkan data menjadi tidak seimbang atau *imbalance*. Adanya ketidakseimbangan antara data di kategori satu dengan kategori lainnya dapat menyebabkan turunnya performansi model dari metode yang digunakan. Idealnya, beberapa metode klasifikasi dalam statistika dapat diterapkan dengan baik pada data yang seimbang. Akan tetapi, apabila metode tersebut dipaksa untuk digunakan pada data yang tidak seimbang, maka nilai prediksi kategori yang didapatkan akan cenderung mengikuti kelas/kategori mayor (kategori yang jumlah datanya lebih besar) (Jiun dan Chen, 2011). Dalam hal ini, pada umumnya kelas mayor akan memiliki nilai akurasi prediksi yang besar, yaitu *sensitivity* yang besar jika kelas mayornya adalah kategori sukses atau *specificity* yang besar jika kelas mayornya adalah kategori gagal. Oleh karena itu, sebelum dilakukan analisis menggunakan metode klasifikasi, perlu dilakukan identifikasi mengenai raio imbalance data. Salah satu penialaian resio imbalance data adalah dengan menggunakan indeks *Imbalance Ratio* ($I_r$). Besarnya indeks $I_r$ dapat dihitung dengan menggunakan rumus sesuai dengan Persamaan (2.27).

$$I_r = \frac{N_c - 1}{N_c} \sum_{i=1}^{N_c} \frac{I_i}{I_n - I_i} \tag{2.27}$$

dengan notasi $c$ menunjukkan indeks kategori dan $N_c$ menunjukkan banyaknya kategori (Tanwani dan Farooq, 2009). Indeks $I_r$ memiliki nilai antara $1 \leq I_r \leq \infty$. Interpretasi nilai yang dihasilkan oleh indeks $I_r$ dituliskan dalam Tabel 2.4 (Fan, dkk., 2016).

**Tabel 2.4** Interpretasi Nilai Indeks $I_r$

| Nilai $I_r$ | Interpretasi |
|---|---|
| $I_r \leq 9$ | Data memiliki sifat *imbalance* yang rendah (*low imbalance*) |
| $9 \leq I_r \leq 20$ | Data memiliki sifat *imbalance* yang sedang (*medium imbalance*) |
| $I_r > 20$ | Data memiliki sifat *imbalance* yang tinggi (*high imbalance*) |

# BAB 3
# METODE PENELITIAN

Bagian ini menjelaskan metode penelitian yang digunakan untuk mencapai tujuan penelitian. Penjelasan mengenai sumber data, baik data simulasi maupun data riil dibahas dalam Subbab 3.1. Sedangkan langkah analisis secara detail yang dilakukan untuk mendapatkan hasil keluaran Lorens dan ELR dijelaskan dalam Subbab 3.2.

## 3.1 Sumber Data

Terdapat dua jenis data yang digunakan untuk mencapai tujuan penelitian dalam analisis ini, yaitu data simulasi dan data riil. Data simulasi digunakan untuk mengetahui performansi metode Lorens dan ELR pada beberapa kasus yang mungkin terjadi, seperti kasus data *balance* dan *imbalance* dalam data berdimensi tinggi, sedangkan analisis pada data riil digunakan untuk mengetahui performansi metode pada suatu studi kasus tertentu. Studi kasus dalam penelitian ini dikhususkan dalam permasalahan *drug discovery*. Penjelasan mengenai kedua jenis data tersebut dijelaskan dalam uraian berikut ini.

### 4.2.1 Data Simulasi

Data yang digunakan untuk studi simulasi didapatkan dari hasil pembangkitan data secara random dengan skenario tertentu. Pada studi simulasi ini, analisis difokuskan untuk mengetahui efek-efek tertentu pada data, seperti efek penambahan banyaknya variabel, efek bertambahnya rasio keseimbangan (*imbalance*) suatu data, dan efek multikolinearitas. Data simulasi diberi kode dengan format “BMV-123”. Kode dengan huruf “B” menunjukkan simulasi yang fokus pada sifat keseimbangan data. Variasi skenario yang dibuat pada simulasi ini ditunjukkan oleh kode angka pertama. Kode dengan huruf “M” menunjukkan simulasi yang fokus pada sifat multikolinieritas data. Variasi skenario yang dibuat pada simulasi ini ditunjukkan oleh kode angka kedua. Sedangkan kode dengan huruf “V” menunjukkan simulasi yang fokus pada sifat penambahan variabel pada data. Variasi skenario yang dibuat pada simulasi ini ditunjukkan oleh kode angka ketiga.

**Tabel 3.1** Skenario Data Simulasi untuk Mengetahui Efek Penambahan Jumlah Variabel

| Kode Skenario | Perbandingan $n{:}p$ | Banyaknya Variabel ($p$) | |
|---|---|---|---|
| | | Diskrit ($p_{diskrit}$) | Kontinyu ($p_{kontinyu}$) |
| BMV-111 | 1:2 | 40 | 160 |
| BMV-112 | 1:3 | 60 | 240 |
| BMV-113 | 1:4 | 80 | 320 |
| BMV-114 | 1:5 | 100 | 400 |
| BMV-115 | 1:6 | 120 | 480 |
| BMV-116 | 1:7 | 140 | 560 |
| BMV-117 | 1:8 | 160 | 640 |
| BMV-118 | 1:9 | 180 | 720 |
| BMV-119 | 1:10 | 200 | 800 |

Skenario untuk mengetahui efek penambahan banyaknya variabel dirancang berdasarkan rasio $n$ dan $p$. Perbandingan $n{:}p$ yang dicobakan dalam analisis ini adalah 1:2, 1:3, ..., 1:10. Jika banyaknya data ($n$) yang dibangkitkan sebanyak 100, maka banyaknya variabel yang digunakan adalah sebanyak 200, 300, ..., 1000. Variabel yang dibangkitkan ini dibagi menjadi dua bagian, yaitu variabel diskrit dan kontinyu dengan perbandingan 1:4. Variabel diskrit dibangkitkan sesuai dengan distribusi binomial dengan parameter $\theta$ sebesar 0,8, sedangkan variabel kontinyu pada skenario ini dibangkitkan berdasarkan distribusi normal dengan parameter $\mu = 5$ dan $\sigma^2 = 0{,}8$. Dengan demikian, hasil analisis pada skenario ini mewakili data yang bersifat univariat dan seimbang (*balance*) dengan ukuran dimensi data yang semakin membesar. Banyaknya variabel diskrit dan kontinyu pada masing-masing skenario perbandingan $n{:}p$ ditunjukkan secara detail pada Tabel 3.1.

**Tabel 3.2** Skenario Data Simulasi untuk Mengetahui Efek Keseimbangan Data

| Kode Skenario | Perbandingan | $I_r$ | Rasio *Imbalance Data* | Jumlah Data | |
|---|---|---|---|---|---|
| | | | | Positif | Negatif |
| BMV-111 | 1:1 | 1 | Rendah | 457 | 443 |
| BMV-211 | 1:20 | 9,14 | Sedang | 60 | 940 |
| BMV-311 | 1:100 | 49,51 | Tinggi | 12 | 988 |

Skenario untuk mengetahui efek keseimbangan data dilakukan dengan mengaplikasikan metode Lorens dan ELR pada data yang dirancang untuk memiliki

tingkat keseimbangan yang berbeda-beda. Dalam analisis ini, rasio kategori negatif dibuat semakin membesar, yaitu 1:1, 1:20, dan 1:100 dengan $n_{neg} > n_{pos}$. Jika data yang dibangkitkan adalah sebanyak 1000, maka perbandingan banyaknya data yang masuk dalam kategori positif dan negatif secara detail ditunjukkan dalam Tabel 3.2. Indeks $I_r$ menunjukkan indeks *imbalance rasio* suatu data. Pada skenario ini, banyaknya variabel yang digunakan adalah 2000, sehingga dapat dianggap mewakili skenario BMV-111 (yaitu perbandingan $n: p = 1: 2$) dengan besar rasio kategori yang berbeda-beda. Sifat multikolinearitas dalam skenario ini diabaikan, sehingga data kontinyu seluruhnya dibangkitkan sesuai dengan distribusi normal dengan parameter yang sama dengan skenario efek penambahan jumlah variabel.

**Tabel 3.3** Skenario Data Simulasi untuk Mengetahui Efek Multikolinearitas Data

| Kode Skenario | Keterangan | Variabel Kontinyu |
|---|---|---|
| BMV-111 | Tidak ada multikolinearitas | Untuk $j = 1, 2, \dots, p$,<br>$x_j \sim N(\mu = 5, \sigma^2 = 0{,}8)$ |
| BMV-121 | Ada multikolinearitas | $\boldsymbol{X} \sim N_{p_{kontinyu}}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$<br>dengan<br>$\boldsymbol{\mu}$ berukuran $p_{kontinyu} \times 1$ dengan $\boldsymbol{\mu} = (5{,}5, \dots, 5)^T$<br>$\boldsymbol{\Sigma}$ berukuran $p_{kontinyu} \times p_{kontinyu}$<br>dengan $\boldsymbol{\Sigma} = \begin{pmatrix} 1 & 0{,}8 & \dots & 0{,}8 \\ 0{,}8 & 1 & \dots & 0{,}8 \\ \vdots & \vdots & \ddots & \vdots \\ 0{,}8 & 0{,}8 & \dots & 1 \end{pmatrix}$ |
| BMV-131 | | Untuk $j = 3, 4, \dots, p$,<br>$x_j \sim N(\mu = 5, \sigma^2 = 0{,}8)$<br>Untuk $j = 1$ dan 2,<br>$x_j = 0{,}7x_{j+2} + 0{,}9x_{j+3}$ |

Analisis simulasi untuk mengetahui efek sifat multikolinearitas data dilakukan terhadap tiga jenis data simulasi. Pertama, data yang tidak memiliki sifat multikolinearitas. Data ini sesuai dengan skenario BMV-111. Kedua, data yang memiliki sifat multikolinearitas yang dihasilkan dari pembagkitan data dengan distribusi normal multivariat. Ketiga, data yang memiliki sifat multikolinearitas yang dihasilkan dari hasil kombinasi linier beberapa variabel tertentu yang berdistribusi normal. Banyaknya data yang dibangkitkan adalah 100 dan banyaknya variabel yang dibangkitkan adalah 200 dengan 160 variabel kontinyu dan 40

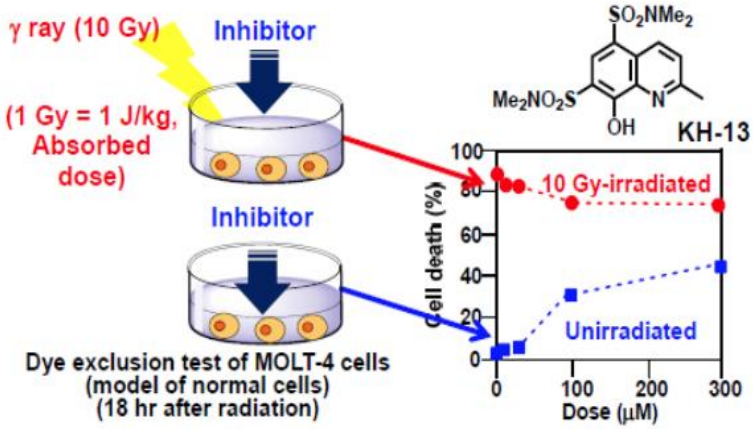
variabel diskrit. Ketiga dataset dirancang agar memiliki rasio yang seimbang pada kedua kategorinya. Variabel diskrit pada analisis ini dibangkitkan dengan distribusi binomial dengan parameter $\theta = 0{,}8$. Tabel 3.3 menunjukkan detail skenario pada analisis data simulasi ini.

Variabel respon didapatkan dari hasil perhitungan berdasarkan Persamaan (2.2). Pada persamaan tersebut, nilai-nilai dalam vektor $\boldsymbol{x}_i$ yang digunakan adalah dari data hasil bangkitan pada uraian yang telah dijelaskan di atas, sedangkan parameter $\beta_0$ dan $\boldsymbol{\beta}$ dibangkitkan secara random berdasarkan distribusi normal dengan parameter $\mu = 0$ dan $\sigma \sim N(5{,}1)$. Data dalam variabel respon ditentukan dengan cara: jika hasil perhitungan nilai $P(Y_i = 1|\boldsymbol{x}_i) > threshold$, maka $Y_i = 1$, dan sebaliknya pada kategori $-1$. Nilai *threshold* ditentukan berdasarkan rasio kelas negatif terhadap total jumlah data yang akan dibangkitkan, misalnya jika perbandingan kelas positif dan negatif yang dibangkitkan adalah 1:3, maka *threshold* yang digunakan adalah 0,75.

### 4.2.2 Data Riil

Pada penelitian ini dilakukan analisis dengan menggunakan data yang dihasilkan dari percobaan yang telah dilakukan oleh Ariyasu, dkk. (2014), yaitu data mengenai proteksi radiasi sel. Banyaknya data dan variabel adalah 84 senyawa (sebagai data) dan 217 karakteristik sel (sebagai variabel). Sel yang digunakan dalam percobaan ini adalah sel MOLT-4 dengan inhibitor p53. Sel ini dipilih karena banyak digunakan dalam percobaan serupa dan juga karena efek inhibitor p53 pada sel tersebut bekerja secara normal.

Percobaan untuk mengamati proteksi radiasi sel dilakukan dengan cara memasukkan senyawa ke dalam sel yang telah diberikan inhibitor p53 kemudian diberikan radiasi sinar $\gamma$ dengan dosis sebesar 10 Gy (1 Gy=1 J/kg untuk dosis radiasi sel). Dalam prosesnya, konsentrasi atau dosis senyawa yang dimasukkan ditingkatkan dari 0 μM hingga 300 μM, kemudian tingkat kematian sel dihitung pada setiap penambahan konsentrasi tersebut. Proses percobaan untuk mengamati toksisitas sel dilakukan dengan cara serupa tanpa melakukan radiasi sinar $\gamma$. Secara singkat kedua proses percobaan ini ditunjukkan dalam Gambar 3.1.



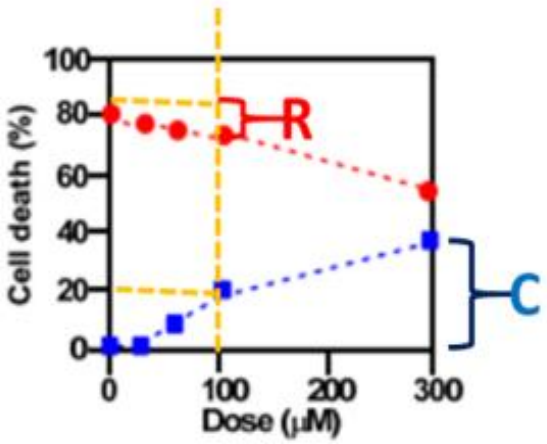

**Gambar 3.1** Contoh Visualisasi Langkah Percobaan untuk Pengamatan Toksisitas dan Proteksi Radiasi Sel oleh Senyawa KH-13 (Ariyasu, dkk., 2014)

Pemberian label data pada pengamatan proteksi radiasi sel (yang selanjutnya digunakan sebagai variabel respon) dilakukan berdasarkan tingkat kematian sel (*cell death rate*) dan bergantung pada nilai *threshold* dari pengamatan toksisitas sel. Pada penelitian sebelumnya, *threshold* data pada pengamatan toksisitas sel yang didapatkan adalah sebesar 20%, sehingga perhitungan nilai *threshold* untuk pengamatan proteksi radiasi sel dapat dituliskan seperti pada Persamaan (3.1).

$$R = cdr_r\ (dosis = 0\ \mu M) - cdr_r\ \big(dosis(cdr_c = 20\%)\big) \tag{3.1}$$

dengan $R$ menunjukkan *threshold* untuk menentukan label proteksi radiasi sel, sedangkan $cdr_c$ dan $cdr_r$ adalah *cell death rate* pada pengamatan pengamatan toksisitas dan proteksi radiasi sel. Contoh gambaran visual Persamaan (3.1) ditunjukkan dalam Gambar 3.2. Perhitungan *threshold* pada pengamatan proteksi radiasi sel yang telah dilakukan pada penelitian sebelumnya adalah sebesar 10% (Matsumoto, dkk., 2015).

Berdasarkan hasil yang didapatkan, label data dapat ditentukan menggunakan aturan berikut. Jika tingkat kematian sel lebih tinggi daripada *threshold*, maka senyawa dikategorikan dalam kategori positif, artinya senyawa tersebut memiliki efek proteksi radiasi yang tinggi. Namun, jika tingkat kematian sel lebih rendah daripada *threshold*, maka senyawa dikategorikan dalam kategori negatif, artinya senyawa tersebut memiliki efek proteksi radiasi yang rendah (Kimura, dkk., 2017).




**Gambar 3.2** Contoh Visualisasi Proses Perhitungan *Threshold* untuk Menentukan Label Data (Kimura, dkk., 2017)

Variabel prediktor yang digunakan adalah variabel yang memuat informasi tentang karakteristik sel, sedangkan variabel responnya adalah mengenai proteksi radiasi sel. Banyaknya karakteristik yang diamati sebagai variabel prediktor adalah 217 variabel yang berkaitan dengan hidrofobisitas sel, struktur sel, ukuran dan berat sel, dan lain-lain (Kimura, dkk., 2017) dan kategori pada variabel respon yang digunakan didefinisikan sebagai kategori positif (kategori 1) dan negatif (kategori -1). Data dalam dataset yang digunakan secara keseluruhan berskala rasio, baik untuk data yang berjenis diskrit maupun kontinyu. Daftar karakteristik yang diamati ditunjukkan dalam Tabel 3.4.

**Tabel 3.4** Karakteristik Sel yang Diamati

| No | Karakteristik Sel | Skala | No | Karakteristik Sel | Skala |
|---|---|---|---|---|---|
| 1 | pKa | Rasio | 12 | ALogP MR | Rasio |
| 2 | Br *Count* | Rasio | ⋮ | ⋮ | ⋮ |
| 3 | C *Count* | Rasio | 160 | *Dipole* mag | Rasio |
| 4 | Cl *Count* | Rasio | 161 | *Dipole* X | Rasio |
| 5 | F *Count* | Rasio | 162 | *Dipole* Y | Rasio |
| 6 | H *Count* | Rasio | 163 | *Dipole* Z | Rasio |
| 7 | I *Count* | Rasio | ⋮ | ⋮ | ⋮ |
| 8 | N *Count* | Rasio | 214 | *Molecular* 3D PolarSASA | Rasio |
| 9 | O *Count* | Rasio | 215 | *Molecular* 3D SASA | Rasio |
| 10 | S *Count* | Rasio | 216 | *Molecular* 3D SAVol | Rasio |
| 11 | ALogP98 | Rasio | 217 | *Molecular Volume* | Rasio |

Data yang digunakan dalam penelitian ini tidak menunjukkan adanya sifat *imbalanced data*, karena jumlah atau rasio data pada masing-masing kategori untuk dataset yang digunakan tidak terlalu memiliki perbedaan atau dapat dianggap sama. Jumlah data yang termasuk dalam kategori positif adalah sebanyak 39 senyawa, sedangkan yang termasuk dalam kategori negatif adalah sebanyak 45 data. Analisis klasifikasi baik pada data simulasi maupun pada data *drug discovery* dilakukan dengan melakukan pembagian partisi data *training* dan *testing* dengan rasio 80:20 secara *stratified*, yaitu proporsional antara kategori positif dan negatifnya. Struktur data pada pengamatan proteksi radiasi sel dalam penelitian ini ditampilkan dalam Tabel 3.5.

**Tabel 3.5** Struktur Data Penelitian

| No | Nama Senyawa | Karakteristik Sel | | | | | | Respon Proteksi Radiasi |
|---|---|---|---|---|---|---|---|---|
| | | pKa | Br *Count* | ⋯ | CIC | ⋯ | *Molecular Volume* | |
| 1 | AS-1 | $x_{1,1}$ | $x_{3,2}$ | ⋯ | $x_{1,j}$ | ⋯ | $x_{3,217}$ | $y_1$ |
| 2 | AS-10 | $x_{2,1}$ | $x_{3,2}$ | ⋯ | $x_{3,j}$ | ⋯ | $x_{3,217}$ | $y_2$ |
| 3 | AS-11 | $x_{3,1}$ | $x_{3,2}$ | ⋯ | $x_{3,j}$ | ⋯ | $x_{3,217}$ | $y_3$ |
| 4 | AS-12 | $x_{4,1}$ | $x_{4,2}$ | ⋯ | $x_{4,j}$ | ⋯ | $x_{4,217}$ | $y_4$ |
| 5 | AS-13 | $x_{5,1}$ | $x_{5,2}$ | ⋯ | $x_{5,j}$ | ⋯ | $x_{5,217}$ | $y_5$ |
| ⋮ | ⋮ | ⋮ | ⋮ | | ⋮ | | ⋮ | ⋮ |
| $i$ | *Naphthalene* | $x_{i,1}$ | $x_{i,2}$ | ⋯ | $x_{i,j}$ | ⋯ | $x_{i,217}$ | $y_i$ |
| ⋮ | ⋮ | ⋮ | ⋮ | | ⋮ | | ⋮ | ⋮ |
| 84 | YT-1 | $x_{84,1}$ | $x_{84,2}$ | ⋯ | $x_{84,j}$ | ⋯ | $x_{84,217}$ | $y_{84}$ |

## 3.2 Langkah Analisis

Berdasarkan rumusan masalah yang telah dijelaskan dan untuk mencapai tujuan penelitian yang disebutkan dalam penjelasan pada Subbab 1.2 dan 1.3, maka dibuatlah tahapan penelitian yang didesain untuk proses analisis dengan menggunakan metode ELR dan Lorens serta untuk membandingkan kedua metode tersebut. Tahapan analisis secara keseluruhan dalam analisis ini dituliskan secara bertahap pada langkah-langkah berikut.

1. Tahapan studi simulasi dan analisis yang dilakukan
   a. Membangkitkan data simulasi dengan skenario sesuai dengan penjelasan pada Subbab 3.1.1

b. Melakukan analisis dengan metode Lorens dan ELR dengan melakukan iterasi sebanyak 10 kali pada masing-masing skenario
c. Menghitung nilai rata-rata performansi model yang dihasilkan pada Langkah 1b dengan menggunakan kriteria akurasi total, BCR, dan AUC
d. Membandingkan hasil analisis dengan menggunakan hasil perhitungan pada Langkah 1c
e. Menarik kesimpulan

2. Tahapan analisis pada data riil
   a. Melakukan eksplorasi data
   b. Melakukan analisis dengan metode Lorens dan ELR dengan melakukan iterasi atau perulangan sebanyak 10 kali pada masing-masing metode
   c. Menghitung nilai rata-rata performansi model, yaitu akurasi total, BCR, dan AUC, yang dihasilkan pada Langkah 2b
   d. Melakukan perbandingan hasil analisis dari metode Lorens dan ELR berdasarkan kriteria nilai rata-rata total akurasi, BCR, dan AUC
   e. Menarik kesimpulan

Kronologi tahapan analisis dengan metode Lorens dan ELR dituliskan dalam langkah-langkah berikut ini.

1. Tahapan analisis dengan menggunakan metode Lorens

   Bagian ini merupakan langkah awal untuk menentukan nilai yang digunakan sebagai input Lorens.
   a. Menentukan banyaknya *subspace*, yaitu sebanyak $m$ *subspace*. Dalam penelitian ini, banyaknya *subspace* yang digunakan adalah 5, 10, 15, ..., 70
   b. Menghitung nilai $\bar{y}$ dan menghitung nilai $threshold$ berdasarkan Persamaan (2.18)
   c. Menentukan banyaknya *ensemble* ($q$). Dalam penelitian ini ditentukan banyaknya $q$ adalah 11 *ensemble*

   Bagian ini menjelaskan rangkaian langkah analisis Lorens.
   d. Membagi data ke dalam partisi *training* dan *testing* dengan perbandingan 80:20 secara *stratified* atau proporsional terhadap banyaknya data pada masing-masing kategori

e. Untuk masing-masing *ensemble*, dengan indeks $q = 1,2,3,\dots,11$ dan untuk masing-masing *subspace* dengan indeks $m = 1,2,\dots,14$,
   (i) Membuat partisi variabel kedalam $m$ *subspace* secara random dengan jumlah yang sama dan saling bebas
   (ii) Memodelkan data *training* dengan regresi logistik biner
   (iii) Menghitung nilai prediksi probabilitas dari model yang didapatkan pada Langkah 1e(ii) untuk masing-masing kategori pada data *training*
   (iv) Mengaplikasikan model yang didapatkan pada langkah 1e(ii) untuk data *testing* dan mendapatkan nilai prediksi probabilitasnya
f. Menghitung nilai rata-rata prediksi probabilitas yang didapatkan dari *subspace* ke-$1,2,\dots,14$ untuk satu *ensemble*
g. Mendapatkan prediksi target atau prediksi variabel respon dari hasil Langkah 1f berdasarkan nilai threshold yang didapatkan dari Langkah 1b
h. Melakukan *majority voting* untuk hasil prediksi target pada langkah 1g dengan cara menghitung banyaknya *ensemble* yang memprediksi data ke dalam kategori 0 dan 1 (untuk masing-masing data).
i. Mendapatkan prediksi target berdasarkan langkah 1h dengan berdasarkan vote yang lebih diantara kedua kategori. Hasil dari langkah ini merupakan hasil akhir dari rangkaian langkah analisis Lorens.

2. Tahapan analisis dengan menggunakan metode ELR

   Bagian ini merupakan langkah awal untuk menentukan nilai yang digunakan sebagai input ELR.

   a. Menentukan parameter regularisasi $\lambda$. Pada penelitian ini, langkah ini dilakukan secara bersamaan dengan proses pemodelan atau estimasi parameter, sehingga penentuan $\lambda$ yang optimum bisa dilakukan berdasarkan model yang menghasilkan performansi paling baik dari sekian nilai $\lambda$ yang mungkin. Nilai $\lambda$ yang dicobakan didapatkan secara otomatis dari *function* yang digunakan dalam *software*
   b. Mendapatkan inisialisasi vektor probabilitas dengan cara menghitung nilai $1 - p$-value dari statistik uji $t$-*test ranking* untuk setiap variabel berdasarkan Persamaan (2.19)

c. Mendapatkan inisialisasi nilai $\overline{BCR}_0$. Pada penelitian ini ditentukan $\overline{BCR}_0 = 0{,}5$ dengan alasan bahwa rasio kategori 1 dan -1 pada kedua *dataset* relatif seimbang

Bagian ini menjelaskan rangkaian langkah analisis ELR.

d. Untuk satu rangkaian iterasi dengan indeks $l = 1,2, \ldots,$

(i) Membagi data ke dalam partisi *training* dan *testing* dengan perbandingan 80:20 secara *stratified* atau proporsional terhadap banyaknya data pada masing-masing kategori

(ii) Mendapatkan $n^*$ dari $p$ variabel secara random berdasarkan vektor $\boldsymbol{prob}_{l-1}$. Notasi $n^*$ menunjukkan jumlah data *testing*

(iii) Membuat model regresi logistik biner terregularisasi $L_2$ untuk data *training* menggunakan $n^*$ variabel terpilih

(iv) Mendapatkan kategori prediksi pada data *training* dan menghitung performansi ke-$l$. Hasil dari langkah ini bukan digunakan untuk perhitungan nilai $quality_l$ dan vektor probabilitas yang baru. Akan tetapi, hanya disimpan agar fluktuasi nilai akurasi total, BCR, AUC pada data *training* terekam secara lengkap

(v) Mengaplikasikan model yang didapatkan pada Langkah 2d(iii) pada data *testing*, kemudian mendapatkan kategori prediksinya

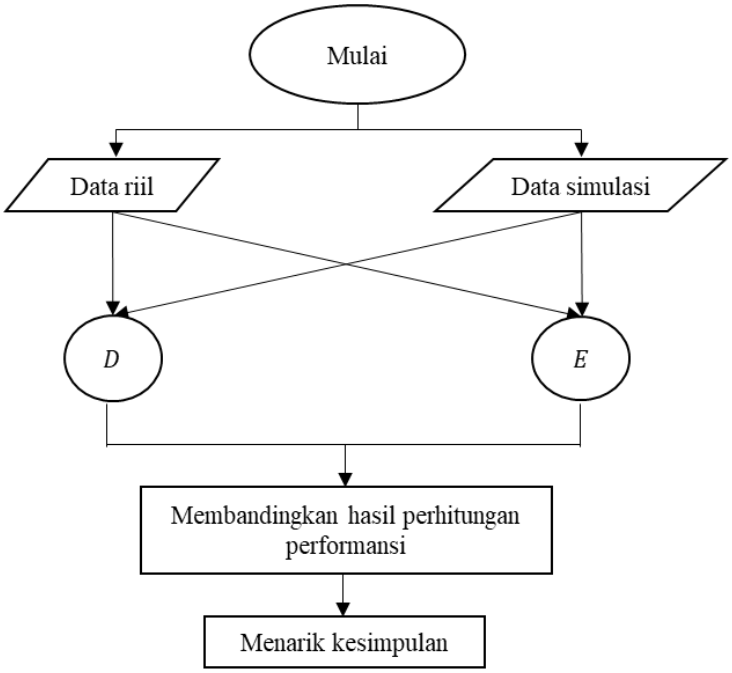
(vi) Menghitung performansi model ke-$l$ pada data *testing* berdasarkan kriteria akurasi total, BCR, dan AUC. Perhitungan nilai akurasi total dan AUC pada langkah ini hanya digunakan sebagai perekaman fluktuasi nilainya saja. Untuk langkah selanjutnya hanya digunakan nilai BCR ke-$l$ sebagai dasar perhitungan

(vii) Menghitung nilai $quality_l$ dari nilai BCR ke-$l$ data *testing* yang didapatkan pada langkah 2d(iv) berdasarkan Persamaan (2.15)

(viii) Memperbarui vektor probabilitas dengan berdasarkan nilai estimasi parameter model yang didapatkan dari Langkah 2d(iii) dan $quality_l$ dari langkah 2d(vii)

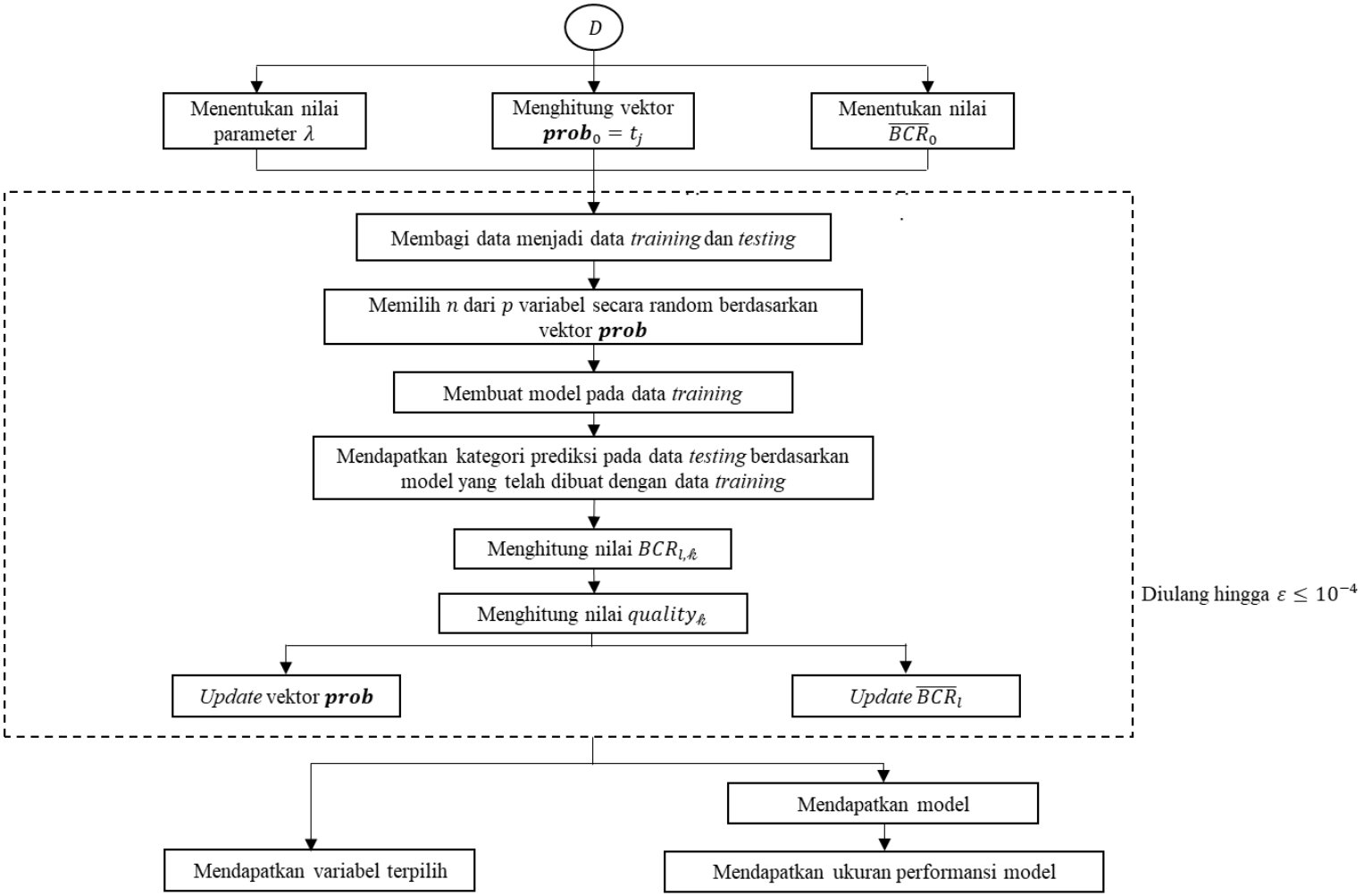
(ix) Menghitung memperbarui nilai $\overline{BCR}_l = \frac{l.\overline{BCR}_{l-1} + BCR_l}{l+1}$

(x) Menghitung nilai $\varepsilon = \overline{BCR}_{l-1} - \overline{BCR}_{l}$. Jika nilai $\varepsilon \leq 10^{-5}$, maka iterasi berhenti, artinya telah didapatkan model yang menghasilkan nilai $\overline{BCR}$ yang konvergen.

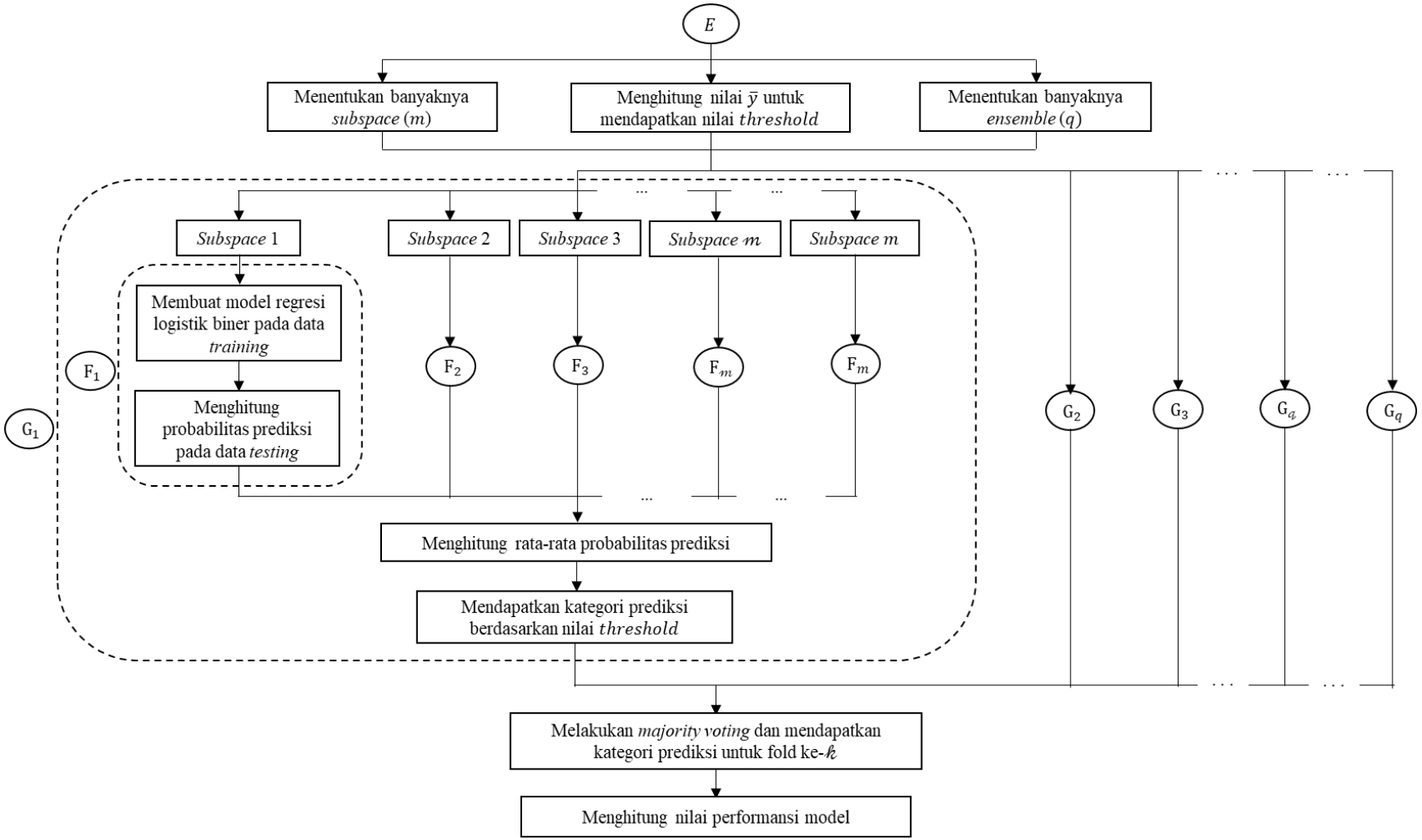



**Gambar 3.3** Diagram Alir Tahapan Penelitian untuk Seluruh Proses Analisis

Langkah D pada Gambar 3.3 merupakan langkah analisis dalam penelitian ini secara keseluruhan. Secara lebih jelas, Langkah D ditunjukkan dalam Gambar 3.4, sedangkan Langkah E ditunjukkan dalam Gambar 3.5. Proses analisis klasifikasi dengan menggunakan *k-fold cross validation* dilakukan terhadap diagram alir pada Gambar 3.4 dan 3.5 dengan disesuaikan pada tahapan penelitian yang telah dijelaskan pada uraian sebelumnya.




**Gambar 3.4** Diagram Alir Analisis Klasifikasi dengan Metode ELR




**Gambar 3.5** Diagram Alir Analisis Klasifikasi dengan metode Lorens

*(Halaman ini sengaja dikosongkan)*

# BAB 4
# ANALISIS DAN PEMBAHASAN

Bagian ini menjelaskan mengenai hasil analisis sesuai dengan rumusan masalah dan tujuan penelitian yang didefinisikan dalam Bab 1. Data yang digunakan dalam analisis ini merupakan data simulasi, yang hasil analisisnya diuraikan dalam Subbab 4.1. Selain itu juga digunakan data riil berupa data *drug discovery* yang berisi informasi mengenai karakteristik proteksi radiasi sel yang hasil analisisnya diuraikan dalam Subbab 4.2.

## 4.1 Penjelasan Algoritma

Algoritma yang ditunjukkan dalam Bab 2 dibahas secara lebih detail dalam bagian ini. Langkah-langkah pada algoritma yang diuraikan dalam pembahasan ini dilakukan untuk mendapatkan model klasifikasi yang hasilnya dibahas pada Subbab 4.2 dan 4.3. Pembahasan mengenai algoritma Lorens dibahas dalam Subbab 4.1.1, sedangkan pembahasan mengenai algoritma ELR dibahas dalam Subbab 4.1.2.

### 4.2.1 *Logistic Regression Ensemble* (Lorens)

Proses *training* atau pelatihan model dalam metode Lorens dilakukan berdasarkan algoritma berikut ini.

**Input**

a. Menentukan data sampel yang akan digunakan dalam analisis. Dimisalkan data sampel dinotasikan dengan $\boldsymbol{X} \in \mathbb{R}^p$ dan variabel respon dinotasikan $\boldsymbol{y} \in \{0,1\}^n$, dengan kategori 0 untuk gagal dan 1 untuk sukses. Elemen matrik $\boldsymbol{X}$ dan $\boldsymbol{y}$ adalah sebagai berikut.

$$\boldsymbol{X} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ x_{31} & x_{32} & \cdots & x_{3p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix} \text{ dan } \boldsymbol{y} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{pmatrix}$$

dengan $n$ merupakan banyaknya data sampel dan $p$ merupakan banyaknya variabel. Jika data yang digunakan merupakan data berdimensi tinggi, maka $n < p$. Jika data dibagi dalam partisi training dan testing, maka pembagian sebaiknya dilakukan secara *stratified* agar data dalam kategori positif dan

negatif terwakili dalam data *training* untuk digunakan sebagai input dalam pembuatan model klasifikasi.

b. Menentukan banyaknya *subspace* ($M$). Penentuan $M$ *subspace* yang digunakan untuk membagi data ditentukan dengan pertimbangan jumlah maksimal variabel yang bisa digunakan dalam satu *subspace* data. Perhitungan ini dilakukan dengan langkah sebagai berikut.
   - Menentukan banyaknya data yang akan digunakan dalam proses pembuatan atau *training* model, dinotasikan dengan $n_{training}$. Jika data dibagi menjadi data *training* dan *testing*, maka digunakan banyaknya data *training* sebagai acuan perhitungan ini.
   - Menentukan jumlah maksimal variabel dalam satu *subspace*. Dengan asumsi bahwa $n \nleq p$, maka jumlah maksimal $p$ yang seharusnya ada didalam satu *subspace* adalah sebanyak $n-1$, sehingga didapatkan banyaknya variabel yang mungkin adalah $1 \leq p_{subspace} \leq n-1$.
   - Mendapatkan banyaknya *subspace* berdasarkan jumlah variabel yang telah ditentukan sesuai dengan pertimbangan pada langkah sebelumnya dengan cara membagi jumlah variabel dengan jumlah maksimal variabel dalam satu *subspace*. Jika didapatkan hasil nilai decimal, sebaiknya dilakukan pembulatan ke atas.Sebagai contoh, misalnya terdapat 100 variabel dalam satu dataset dan banyaknya data adalah 50, maka banyaknya *subspace* yang bisa digunakan dalam analisis ini adalah $\frac{100}{50-1} = 2{,}0408 \approx 3$. Dengan demikian jumlah subspace minimal yang bisa digunakan adalah sebanyak 3, sedangkan jumlah maksimal *subspace* adalah sebanyak jumlah variabel.

c. Menentukan *threshold* yang digunakan untuk menentukan prediksi kategori dari hasil prediksi probabilitas. Biasanya nilai *threshold* yang digunakan adalah 0,5. Akan tetapi, apabila data termasuk ke dalam data yang *imbalance*, maka *threshold* dapat ditentukan berdasarkan Persamaan (2.18).

d. Menentukan banyaknya ensemble yang digunakan sebagai pengulangan serangkaian algoritma Lorens. Banyaknya ensemble dinotasikan $q = 1,2,\dots,Q$.

***Output***

Prediksi kategori $\widehat{\boldsymbol{y}}$

***Begin***

***Do*** $\boldsymbol{q = 1}$ ***to*** **Q**

a. Membagi data ke dalam *M subspace*, sehingga $\boldsymbol{X}_m \in \theta_m$, dengan $\boldsymbol{X}_m$ menunjukkan matrik data yang terdapat dalam subspace ke-$m$ dan $\theta_m$ menunjukkan dimensi data yang terdapat dalam *subspace* ke-$m$. Pembagian data ke dalam *M subspace* juga dilakukan pada data testing.

***Do*** $\boldsymbol{m = 1}$ **to** $\boldsymbol{M}$

b. Memodelkan data *training* yang termasuk dalam subspace ke-$m$ atau $\boldsymbol{X}_m$ dengan regresi logistik biner.

c. Mendapatkan prediksi probabilitas data training atau $\widehat{\boldsymbol{\pi}}_m(\boldsymbol{x})$ berdasarkan model regresi logistik yang didapatkan pada Langkah b.

d. Mendapatkan prediksi probabilitas data testing atau $\widehat{\boldsymbol{\pi}}_m^*(\boldsymbol{x})$ berdasarkan model regresi logistik yang didapatkan pada Langkah b.

***End do***

e. Menghitung nilai rata-rata $\widehat{\boldsymbol{\pi}}(\boldsymbol{x})$ dan $\widehat{\boldsymbol{\pi}}^*(\boldsymbol{x})$ untuk ensemble ke-$q$, dengan $\left\{\overline{\widehat{\boldsymbol{\pi}}}(\boldsymbol{x})\right\}_q = \frac{1}{M}\left(\widehat{\boldsymbol{\pi}}_1(\boldsymbol{x}) + \widehat{\boldsymbol{\pi}}_2(\boldsymbol{x}) + \cdots + \widehat{\boldsymbol{\pi}}_M(\boldsymbol{x})\right) = \frac{1}{M}\sum_{m=1}^{M}\widehat{\boldsymbol{\pi}}_m(\boldsymbol{x})$ dan $\left\{\overline{\widehat{\boldsymbol{\pi}}}^*(\boldsymbol{x})\right\}_q = \frac{1}{M}\left(\widehat{\boldsymbol{\pi}}_1^*(\boldsymbol{x}) + \widehat{\boldsymbol{\pi}}_2^*(\boldsymbol{x}) + \cdots + \widehat{\boldsymbol{\pi}}_M^*(\boldsymbol{x})\right) = \frac{1}{M}\sum_{m=1}^{M}\widehat{\boldsymbol{\pi}}_m^*(\boldsymbol{x})$

dengan *M* adalah banyaknya *subspace*, $\left\{\overline{\widehat{\boldsymbol{\pi}}}(\boldsymbol{x})\right\}_q$ adalah rata-rata prediksi probabilitas untuk ensemble ke-$q$ pada data training, sedangkan $\left\{\overline{\widehat{\boldsymbol{\pi}}}^*(\boldsymbol{x})\right\}_q$ adalah rata-rata prediksi probabilitas untuk ensemble ke-$q$ pada data testing.

f. Mendapatkan prediksi kategori $\widehat{\boldsymbol{y}}_q$ dan $\widehat{\boldsymbol{y}}_q^*$ berdasarkan nilai rata-rata yang didapatkan dari Langkah e berdasarkan nilai *threshold* yang telah ditentukan pada Input c. Notas $\widehat{\boldsymbol{y}}_q$ menunjukkan kategori prediksi pada data training dan $\widehat{\boldsymbol{y}}_q^*$ menunjukkan kategori prediksi pada data testing.

***End do***

g. Melakukan *majority voting* dari output $\widehat{\boldsymbol{y}}_q$ dan $\widehat{\boldsymbol{y}}_q^*$, untuk $q = 1,2,\ldots Q$. Elemen vektor $\widehat{\boldsymbol{y}}_q$ ditunjukkan dalam Tabel 4.1, dengan $\hat{y}_{i,q} = \{0,1\}$ untuk

$i = 1,2, \dots, n$ dan $q = 1,2, \dots, Q$. Elemen vektor $\hat{\boldsymbol{y}}_q^*$ memiliki bentuk yang sama seperti Tabel 4.1. Hasil prediksi kategori paling akhir dari rangkaian algoritma Lorens ini ditentukan berdasarkan Tabel 4.1. Data diklasifikasikan kedalam kategori 0 apabila jumlah vote kategori 0 lebih banyak daripada kategori 1, dan sebaliknya.

**Tabel 4.1** Kategori Prediksi Lorens pada Data *Training*

| Id. | *Ensemble* | | | | | | |
|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | ... | q | ... | Q |
| 1 | $\hat{y}_{1,1}$ | $\hat{y}_{1,2}$ | $\hat{y}_{1,3}$ | ... | $\hat{y}_{1,q}$ | ... | $\hat{y}_{1,Q}$ |
| 2 | $\hat{y}_{2,1}$ | $\hat{y}_{2,2}$ | $\hat{y}_{2,3}$ | ... | $\hat{y}_{2,q}$ | ... | $\hat{y}_{2,Q}$ |
| 3 | $\hat{y}_{3,1}$ | $\hat{y}_{3,2}$ | $\hat{y}_{3,3}$ | ... | $\hat{y}_{3,q}$ | ... | $\hat{y}_{3,Q}$ |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋱ | ⋮ | ⋱ | ⋮ |
| i | $\hat{y}_{i,1}$ | $\hat{y}_{i,2}$ | $\hat{y}_{i,3}$ | ... | $\hat{y}_{i,q}$ | ... | $\hat{y}_{i,Q}$ |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋱ | ⋮ | ⋱ | ⋮ |
| n | $\hat{y}_{n,1}$ | $\hat{y}_{n,2}$ | $\hat{y}_{n,1}$ | ... | $\hat{y}_{n,q}$ | ... | $\hat{y}_{n,Q}$ |

***End***

### 4.2.2 *Ensemble Logistic Regression* (ELR)

Proses *training* atau pelatihan model dalam metode ELR dilakukan berdasarkan algoritma berikut ini.

**Input**

a. Menentukan data sampel yang akan digunakan dalam analisis. Dimisalkan data sampel dinotasikan dengan $\boldsymbol{X} \in \mathbb{R}^p$ dan variabel respon dinotasikan $\boldsymbol{y} \in \{0,1\}^n$, dengan kategori 0 untuk gagal dan 1 untuk sukses. Elemen matrik $\boldsymbol{X}$ dan $\boldsymbol{y}$ adalah sebagai berikut.

$$\boldsymbol{X} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ x_{31} & x_{32} & \cdots & x_{3p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix} \text{ dan } \boldsymbol{y} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{pmatrix}$$

dengan $n$ merupakan banyaknya data sampel dan $p$ merupakan banyaknya variabel. Jika data yang digunakan merupakan data berdimensi tinggi, maka $n < p$.

b. Menentukan parameter regularisasi $\lambda$ sebagai parameter regularisasi $l_2$. Pemilihan nilai untuk $\lambda$ sangat penting untuk dilakukan, karena parameter

$\lambda$ merupakan pembobot untuk unsur regularisasi, seperti pada Persamaan (2.17). Nilai $\lambda$ yang terlalu besar akan menyebabkan model cenderung menjadi underfitting, sedangkan jika nilai $\lambda$ terlalu kecil menyebabkan model cenderung overfitting.

c. Mendapatkan inisial vektor probabilitas yang dihasilkan dari nilai $1-p$-*value* statistik uji $t$ dari $t$-*test ranking.*
d. Menentukan nilai inisial $\overline{BCR}_l$ dengan $l = 0$. Untuk data yang seimbang atau *balance*, dapat digunakan 0,5 sebagai $\overline{BCR}_0$.

***Output***

Model regresi logistik terregularisasi dengan nilai $\overline{BCR}$ yang konvergen. Selain itu, juga bisa didapatkan ukuran performansi model pada saat proses sudah menuju konvergen.

***Begin***

***Repeat***

a. Membuat partisi data menjadi data *training* dan *testing*. Pembagian sebaiknya dilakukan secara *stratified* agar data dalam kategori positif dan negatif terwakili dalam data *training* untuk digunakan sebagai input dalam pembuatan model klasifikasi.
b. Mendapatkan $n^*$ dari $p$ variabel secara random berdasarkan vektor probabilitas yang telah ditentukan pada Input c. Notasi $n^*$ menunjukkan jumlah data *training*. Dengan demikian, data yang digunakan dalam analisis adalah berukuran $n^* \times n^*$. Dalam langkah ini, variabel yang memiliki probabilitas yang besar belum tentu terpilih untuk dijadikan sebagai variabel prediktor pada langkah selanjutnya, tetapi nilai probabilitas menunjukkan besar kemungkinan suatu variabel untuk terpilih. Pada setiap iterasi ELR, variabel yang terpilih memiliki kemungkinan berbeda dari daftar variabel terpilih pada iterasi sebelumnya.
c. Membuat model regresi logistik terregularisasi untuk variabel terpilih. Input parameter $\lambda$ digunakan pada langkah ini. Penentuan nilai $\lambda$ dapat dilakukan bersamaan dengan proses pemodelan pada langkah ini. Jika dipilih pada saat melakukan pemodelan, kriteria nilai $\lambda$ yang sesuai adalah

yang dapat menghasilkan error paling kecil atau ukuran kebaikan model yang besar.

d. Mendapatkan nilai prediksi kategori pada data training dan data testing berdasarkan model yang didapatkan dari Langkah c. Hasil prediksi kategori yang digunakan sebagai input pada langkah selanjutnya adalah prediksi dari data testing, yaitu dinotasikan dengan $\widehat{\boldsymbol{y}}^{*}$.
e. Menghitung nilai $BCR_l$ untuk iterasi ke-$l$ berdasarkan hasil prediksi kategori data testing yang didapatkan dari Langkah d. Untuk memudahkan perhitungan $BCR_l$, biasanya dibuat confusion matrix seperti Tabel 2.2.
f. Menghitung nilai quality berdasarkan $BCR_l$ yang didapatkan pada Langkah e. Pada perhitungan ini, jika $\overline{BCR}_{l-1} > BCR_l$ makaakan dihasilkan nilai $quality < 0$, dan sebaliknya. Tanda dan besarnya nilai *quality* ini memengaruhi hasil pada perhitungan setelahnya.
g. Memperbarui vektor $\boldsymbol{prob}_l$ berdasarkan nilai *quality* yang didapatkan pada Langkah f. Pada penelitian ini, perhitungan update probabilitas dilakukan terlebih dahulu tanpa menggunakan nilai normalisasi $z$, sehingga ada kemungkinan hasil perhitungan yang didapatkan lebih dari 1. Selanjutnya, sebagai pembobot agar hasil update berupa nilai probabilitas yang besarnya adalah diantara 0 dan 1, maka normalisasi $z$ didefinisikan sebagai nilai hasil perhitungan update nilai probabilitas (tanpa pembobot $z$) yang terbesar dari variabel yang terpilih. Nilai hasil perhitungan quality pada langkah ini digunakan sebagai pembobot untuk perubahan nilai probabilitas.
h. Memperbarui nilai $\overline{BCR}_l$ dengan rumusan yang telah didefinisikan pada algoritma ELR

***Until*** $\varepsilon = \overline{BCR}_{l-1} - \overline{BCR}_l$ konvergen

***End***

Besarnya nilai kriteria konvergensi pada algoritma ELR pada penelitian ini yang digunakan adalah $10^{-5}$.

## 4.2 Analisis Data Simulasi

Analisis data simulasi pada penelitian ini difokuskan kedalam 3 hal, yaitu untuk mengetahui efek penambahan jumlah variabel, efek multikolinearitas, dan efek keseimbangan data. Pada analisis menggunakan metode Lorens, efek jumlah partisi juga menjadi salah satu hal yang akan diamati pada bagian ini. Analisis data simulasi menggunakan metode Lorens dan ELR secara lebih detail dibahas dalam Subbab 4.1.1, 4.1.2, dan 4.1.3.

### 4.2.3 Analisis Data Simulasi untuk Mengetahui Efek Banyaknya Jumlah Variabel

Skenario untuk mengetahui efek penambahan jumlah variabel yang digunakan adalah sesuai dengan penjelasan pada Subbab 3.1.1, sedangkan untuk mengetahui efek dari perbedaan banyaknya *subspace*, analisis dilakukan dengan skenario tambahan, yaitu dengan melakukan analisis pada beberapa jenis partisi. Jumlah partisi ini ditentukan secara bertingkat, yaitu 5, 10, 15, ..., 70, Angka ini ditentukan dengan pertimbangan bahwa banyaknya variabel pada setiap *subspace* dalam satu *ensemble* tidak lebih dari banyaknya data. Jika dimisalkan pembagian data *training-testing* yang digunakan adalah 80:20 dan digunakan skenario BMV-111, maka perhitungan jumlah partisi minimalnya adalah sebagai berikut.

$n = 100$ dan $p = 200$

$$n_{training} = \frac{80}{100} \times 100 = 80$$

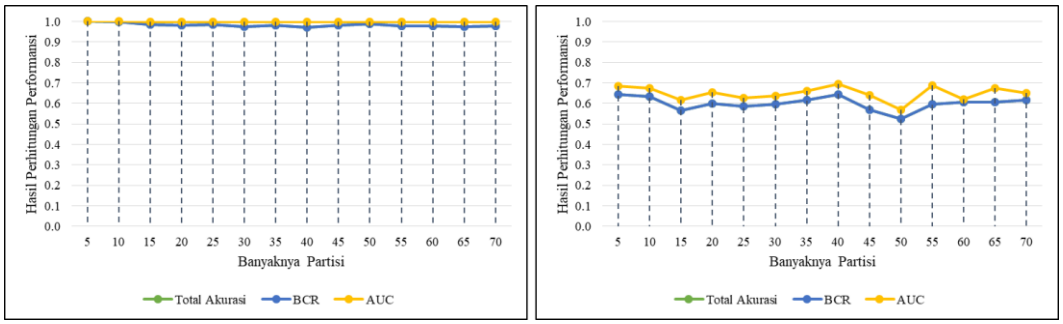
$$p_{subspace} = n_{training} - 1 = 80 - 1 = 79$$

$$m = \frac{200}{79} = 2{,}5316 \approx 3$$

Pada skenario BMV-111 yang digunakan, banyaknya observasi adalah 100 dan banyaknya variabelnya adalah 200, Jika rasio *training-testing* yang digunakan adalah 80:20, maka banyaknya data *training* untuk analisis adalah 80, Dengan asumsi bahwa maksimal banyaknya variabel pada tiap-tiap *subspace* adalah sebanyak data *training*-1, maka didapatkan sebanyak 79 variabel pada tiap-tiap *subspace*. Dengan demikian, minimal jumlah partisi yang seharusnya digunakan adalah 3 partisi (dengan pembulatan ke atas). Dengan pertimbangan ini, maka penentuan jumlah partisi 5, 10, 15, ..., 70 dapat dilakukan. Hasil analisis dengan

menggunakan Lorens akan berubah-ubah sesuai dengan variabel yang terbagi ke dalam beberapa *subspace*. Untuk mengatasi hal ini, maka analisis dilakukan secara berulang sebanyak 10 kali, kemudian hasil akhir ditentukan dengan mengitung nilai rata-rata ukuran performansinya. Gambaran atau ilustrasi secara detail proses perhitungan dalam algoritma Lorens dan ELR dapat dilihat dalam Lampiran 1 dan 2.

Hasil perhitungan nilai rata-rata total akurasi, BCR, dan AUC pada simulasi ini ditunjukkan pada Tabel 4.2 dan divisualisasikan pada Gambar 4.1. Pada Gambar 4.1 dapat dilihat bahwa model Lorens memiliki kecenderungan performansi yang sangat baik dan stabil pada data *training* dengan nilai yang mendekati sempurna pada ketiga ukuran performansi. Akan tetapi, pada data *testing* ketiga performansi mengalami perbedaan yang signifikan jika dibandingkan dengan yang terdapat pada data *training*. Namun, nilai-nilai ini cukup stabil di sekitar angka tertentu, yaitu sekitar 0,6. Hal ini mengindikasikan adanya *overfitting* pada model yang diperoleh, sehingga mengakibatkan performansi model pada data *training* jauh lebih baik daripada data *testing*. Secara lebih rinci, nilai-nilai yang terdapat pada Gambar 4.1 dituliskan dalam Tabel 4.2.




(a) (b)

**Gambar 4.1** Visualisasi Hasil Peritungan Performansi Lorens pada Data Simulasi BMV-111: (a) Performansi Data *Training* dan (b) Performansi Data *Testing*

Tabel 4.2 menunjukkan hasil perhitungan ukuran performansi Lorens pada data simulasi BMV-111 secara lebih rinci. Nilai-nilai yang dicetak tebal menujukkan performansi yang paling tinggi dalam skenario peningkatan jumlah partisi. Dalam tabel tersebut, jika diperhatikan pada data *training*, maka performansi paling baik yang didapatkan adalah 1 atau 100% pada ketiga kriteria

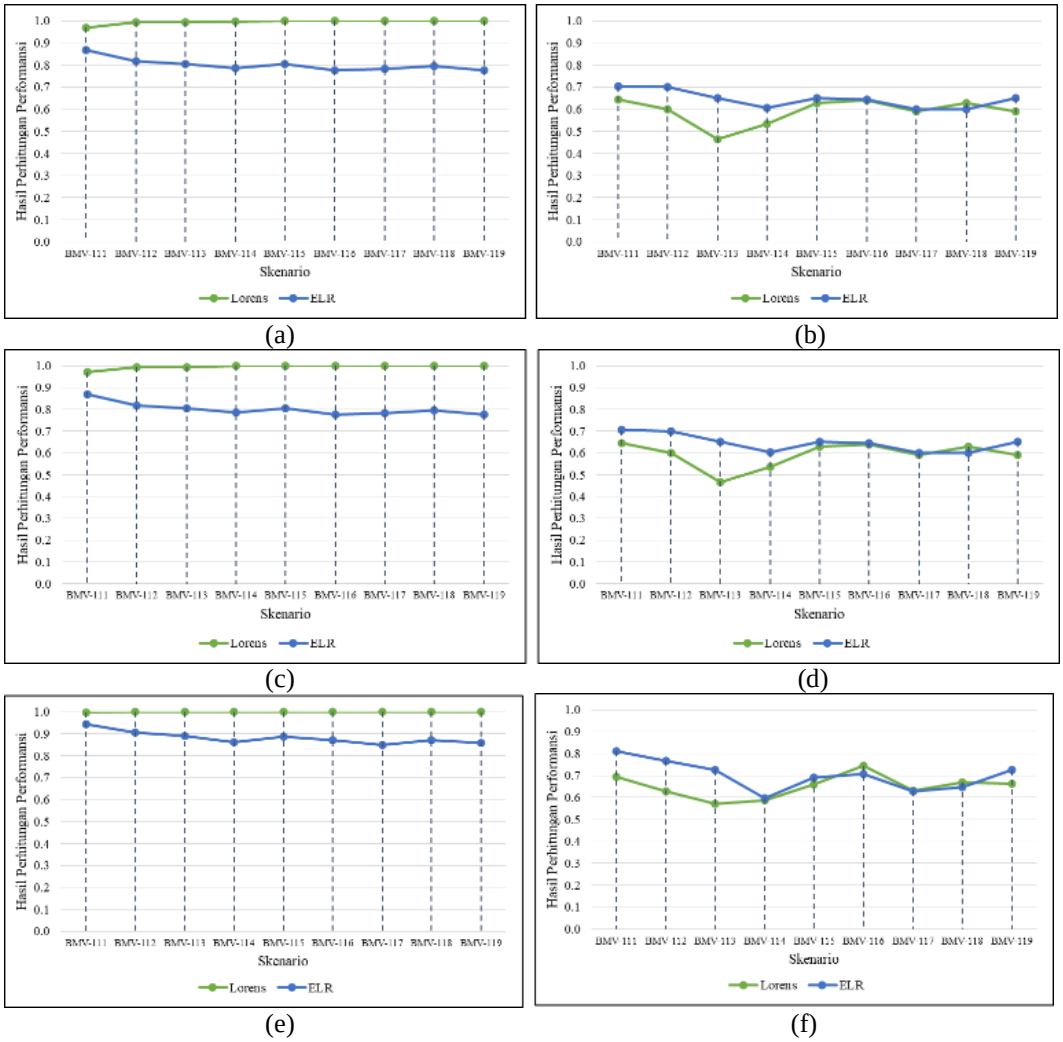
pengukuran, yaitu jika digunakan 5 partisi dalam analisisnya. Akan tetapi, jika model ini diaplikasikan pada data *testing*, performansinya menjadi turun drastis di sekitar nilai 0,6. Pada kolom data *testing*, performansi yang paling tinggi pada kriteria total akurasi dan BCR adalah pada skenario 5 dan 40 partisi. Akan tetapi, hasil perhitungan AUC paling tinggi yang didapatkan adalah hanya pada skenario 40 partisi. Jika model terbaik ditentukan berdasarkan model yang menghasilkan performansi yang sangat baik pada data *testing*, maka skenario 40 partisi merupakan skenario yang terpilih. Dengan demikian, pada pembahasan selanjutnya model Lorens yang digunakan sebagai perbandingan dengan hasil yang didapatkan oleh ELR dalam hal pada skenario yang dibuat untuk mengetahui efek penambahan jumlah variabel adalah model dengan 40 partisi.

**Tabel 4.2** Hasil Perhitungan Nilai Performansi Model Lorens pada Data Simulasi BMV-111

| Partisi | Data *Training* | | | Data *Testing* | | |
|---|---|---|---|---|---|---|
| | Total Akurasi | BCR | AUC | Total Akurasi | BCR | AUC |
| 5 | **1,0000** | **1,0000** | **1,0000** | **0,6450** | **0,6450** | 0,6860 |
| 10 | 0,9975 | 0,9975 | 1,0000 | 0,6350 | 0,6350 | 0,6730 |
| 15 | 0,9850 | 0,9850 | 0,9997 | 0,5650 | 0,5650 | 0,6160 |
| 20 | 0,9813 | 0,9813 | 0,9984 | 0,6000 | 0,6000 | 0,6550 |
| 25 | 0,9838 | 0,9838 | 0,9988 | 0,5850 | 0,5850 | 0,6250 |
| 30 | 0,9725 | 0,9725 | 0,9983 | 0,5950 | 0,5950 | 0,6360 |
| 35 | 0,9813 | 0,9813 | 0,9980 | 0,6150 | 0,6150 | 0,6620 |
| 40 | 0,9700 | 0,9700 | 0,9985 | **0,6450** | **0,6450** | **0,6950** |
| 45 | 0,9813 | 0,9813 | 0,9984 | 0,5700 | 0,5700 | 0,6400 |
| 50 | 0,9888 | 0,9888 | 0,9989 | 0,5250 | 0,5250 | 0,5690 |
| 55 | 0,9788 | 0,9788 | 0,9984 | 0,5950 | 0,5950 | 0,6880 |
| 60 | 0,9788 | 0,9788 | 0,9975 | 0,6050 | 0,6050 | 0,6190 |
| 65 | 0,9738 | 0,9738 | 0,9972 | 0,6050 | 0,6050 | 0,6750 |
| 70 | 0,9775 | 0,9775 | 0,9978 | 0,6150 | 0,6150 | 0,6510 |

Perbandingan performansi Lorens dan ELR pada simulasi mengenai efek penambahan jumlah variabel digambarkan dalam Gambar 4.2. Pada data *training*, penambahan jumlah variabel tidak menunjukkan adanya efek yang signifikan pada performansi Lorens, baik berdasakan nilai total akurasi, BCR, maupun AUC. Gambar 4.2a, 4.2c, dan 4.2e menunjukkan bahwa performansi Lorens pada data

*training* cenderung stabil dan mendekati sempurna. Jika dibandingkan dengan pola performansi yang dihasilkan oleh ELR, Lorens jauh lebih baik. ELR menghasilkan model dengan performansi selalu lebih kecil dari Lorens pada semua skenario simulasi. Skenario penambahan jumlah variabel membuat performansi ELR cenderung menurun. Penurunan terbesar terlihat pada perubahan skenario BMV-111 menjadi BMV-112 atau perbandingan $n:p$ berubah dari 1:2 menjadi 1:3. Pada perubahan skenario selanjutnya performansi ELR cenderung stabil di sekitar 0,8 pada kriteria total akurasi dan BCR, sedangkan kriteria AUC berada di antara nilai 0,8 hingga 0,9.




**Gambar 4.2** Visualisasi Perbandingan Performansi untuk Mengetahui Efek Penambahan Banyaknya Variabel: (a) dan (b) Total Akurasi pada Data *Training* dan *Testing*, (c) dan (d) BCR pada Data *Training* dan *Testing*, serta (e) dan (f) AUC pada Data *Training* dan *Testing*

Perbandingan performansi Lorens dan ELR pada data *testing* menunjukkan hal yang cukup berbeda jika dibandingkan dengan perbandingan pada data *training*. ELR cenderung menghasilkan performansi yang lebih baik pada beberapa skenario. Berdasarkan kriteria total akurasi dan BCR, ELR menghasilkan performansi yang lebih baik hingga skenario BMV-115 daripada Lorens, walaupun pola penurunan terjadi pada 4 skenario pertama. Pola penurunan performansi ELR yang ditunjukkan pada Gambar 4.2b dan 4.2d tidak sebesar Lorens, sehingga pada saat Lorens mulai menunjukkan pola kenaikan, hasil performansinya masih tidak melebihi performansi ELR.

Sedangkan pada skenario selanjutnya, yaitu BMV-116 hingga BMV-119, performansi kedua metode memiliki pola yang cukup random dan tidak jauh berbeda. Perubahan performansi yang ditunjukkan berdasarkan nilai AUC pada Gambar 4.2f menunjukkan pola yang hampir sama dengan dua kriteria lainnya. Akan tetapi pada skenario BMV-114, Lorens dan ELR menghasilkan performansi yang hampir sama. Jika hasil analisis pada data *training* dibandingkan dengan data *testing*, maka dapat ditunjukkan bahwa perbedaan performansi Lorens lebih besar dari pada ELR. Hal ini mengindikasikan bahwa Lorens cenderung menghasilkan model yang overfitting terhadap data. Oleh karena itu, berdasarkan analisis pada data *testing*, ELR dianggap lebih baik daripada Lorens walaupun pada data *training* menunjukkan kecenderungan yang berbeda.

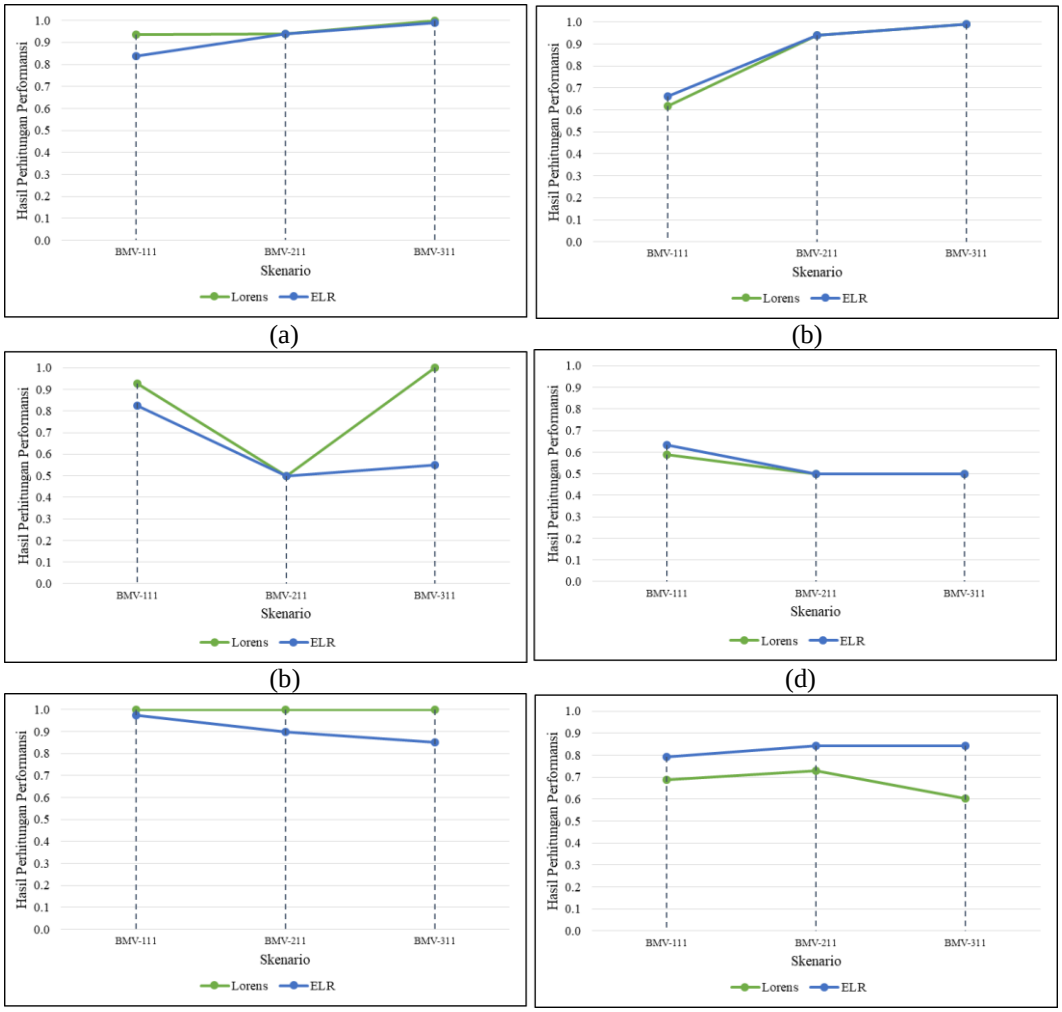
### 4.2.4 Analisis Data Simulasi untuk Mengetahui Efek Keseimbangan Data

Analisis data simulasi pada bagian ini fokus untuk mengetahui efek keseimbangan data. Langkah awal yang dilakukan sebelum mengamati hasil analisis pada setiap skenario adalah menentukan jumlah partisi. Skenario yang digunakan untuk mendapatkan jumlah partisi yang optimum adalah sama dengan yang digunakan dalam analisis simulasi sebelumnya, yaitu BMV-111. Oleh karena itu, dalam uraian ini jumlah partisi yang digunakan disesuaikan dengan hasi analisis dan sesuai dengan penjelasan pada Subbab 4.2.1, yaitu 40 partisi.

Efek keseimbangan data pada metode Lorens dan ELR yang ditunjukkan dalam Gambar 4.3a, 4.3c, dan 4.3e memiliki kecenderungan yang berbeda. Dengan berdasarkan nilai total akurasi, semakin tinggi tingkat ketidakseimbangan data

menyebabkan performansinya semakin baik, baik pada Lorens maupun ELR. Hal ini bisa dikarenakan sifat total akurasi yang kurang sensitif terhadap kasus data *imbalance* atau tidak seimbang, sehingga menghasilkan nilai yang cenderung semakin naik. Dengan berdasarkan nilai BCR, pola performansi Lorens dan ELR cenderung agak membingungkan, karena jika metode tersebut diaplikasikan pada data yang memiliki sifat ketidakseimbangan sedang ($9 < I_r \leq 20$), performansinya turun secara signifikan. Akan tetapi, pada saat diaplikasikan pada data yang memiliki sifat ketidakseimbangan tinggi ($I_r > 20$), Lorens cenderung meningkat secara drastis hingga mendekati 1, sedangkan ELR cenderung meningkat tetapi tidak terlalu besar selisihnya. Pola yang ditunjukkan berdasarkan nilai AUC adalah cenderung semakin menurun untuk hasil ELR dan cenderung stabil untuk Lorens. Walaupun ketiga kriteria kebaikan data menunjukkan pola yang berbeda pada data *training* ini, secara garis besar dapat dilihat bahwa Lorens cenderung memiliki nilai yang selalu lebih tinggi daripada ELR.

Hasil yang ditunjukkan pada data *testing* cukup berbeda jika dibandingkan dengan hasil pada data *training*. Pada Gambar 4.3b, naiknya nilai ketidakseimbangan data menyebabkan performansi Lorens dan ELR cenderung naik. Hal ini disebabkan oleh alasan yang sama dengan pembahasan pada data *training*, yaitu dimungkinkan karena ketidaksensitifan total akurasi untuk menilai performansi model pada data yang tidak seimbang. Pada Gambar 4.3d, pola performansi Lorens dan ELR cenderung menurun jika nilai ketidakseimbangan data semakin ditingkatkan. Akan tetapi perbedaan nilai BCR yang ditunjukkan antara Lorens dan ELR tidak jauh berbeda. Sedangkan berdasarkan nilai AUC yang ditunjukkan pada Gambar 4.3f, pola yang dihasilkan tidak konsisten. ELR menghasilkan nilai AUC yang cenderung naik tetapi tidak signifikan, sedangkan Lorens menghasilkan performansi yang sedikit naik pada skenario kedua kemudian turun secara drastis pada skenario ketiga. Walaupun pola performansi yang ditunjukkan oleh ketiga gambar cukup berbeda, ketiga kriteria kebaikan model tersebut menunjukkan bahwa ELR cenderung lebih baik daripada Lorens, khususnya jika didasarkan pada nilai AUC.



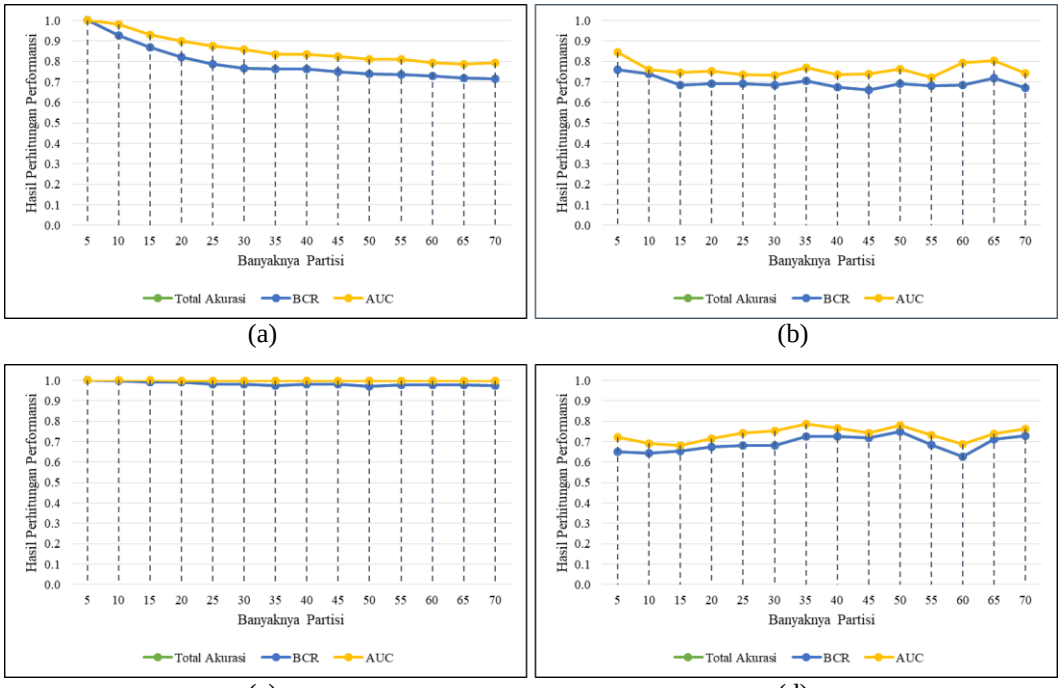

**Gambar 4.3** Visualisasi Perbandingan Performansi untuk Mengetahui Efek Keseimbangan Data: (a) dan (b) Total Akurasi pada Data *Training* dan *Testing*, (c) dan (d) BCR pada Data *Training* dan *Testing*, serta (e) dan (f) AUC pada Data *Training* dan *Testing*

### 4.2.5 Analisis Data Simulasi untuk Mengetahui Efek Multikolinearitas

Analisis data simulasi pada bagian ini bertujuan untuk mengetahui performansi penerapan metode Lorens dan ELR pada data yang memiliki sifat multikolinearitas. Data yang dibangkitkan dalam analisis ini ada sebanyak tiga jenis, yaitu data yang tidak memiliki sifat multikolinearitas (BMV-111), data yang memiliki sifat multikolinearitas dari distribusi normal multivariat (BMV-121), dan data yang memiliki sifat multikolinearitas dari suatu kombinasi linier (BMV-131).

Metode Lorens dalam analisis ini dilakukan dengan langkah yang sama dengan pembahasan pada bagian sebelumnya, yaitu diawali dengan pemilihan banyaknya partisi yang menghasilkan performansi paling baik. Agar perbandingan analisis yang dilakukan menggunakan data yang seragam, maka penentuan jumlah

partisi optimum disesuaikan dengan hasil analisis pada Subbab 4.2.1, yaitu sebanyak 40 partisi.




**Gambar 4.4** Visualisasi Performansi Lorens untuk Data Simulasi: (a) Performansi Data *Training* pada Skenario BMV-121, (b) Performansi Data *Testing* pada Skenario BMV-121, (c) Performansi Data *Training* pada Skenario BMV-131, dan (d) Performansi Data *Testing* pada Skenario BMV-131

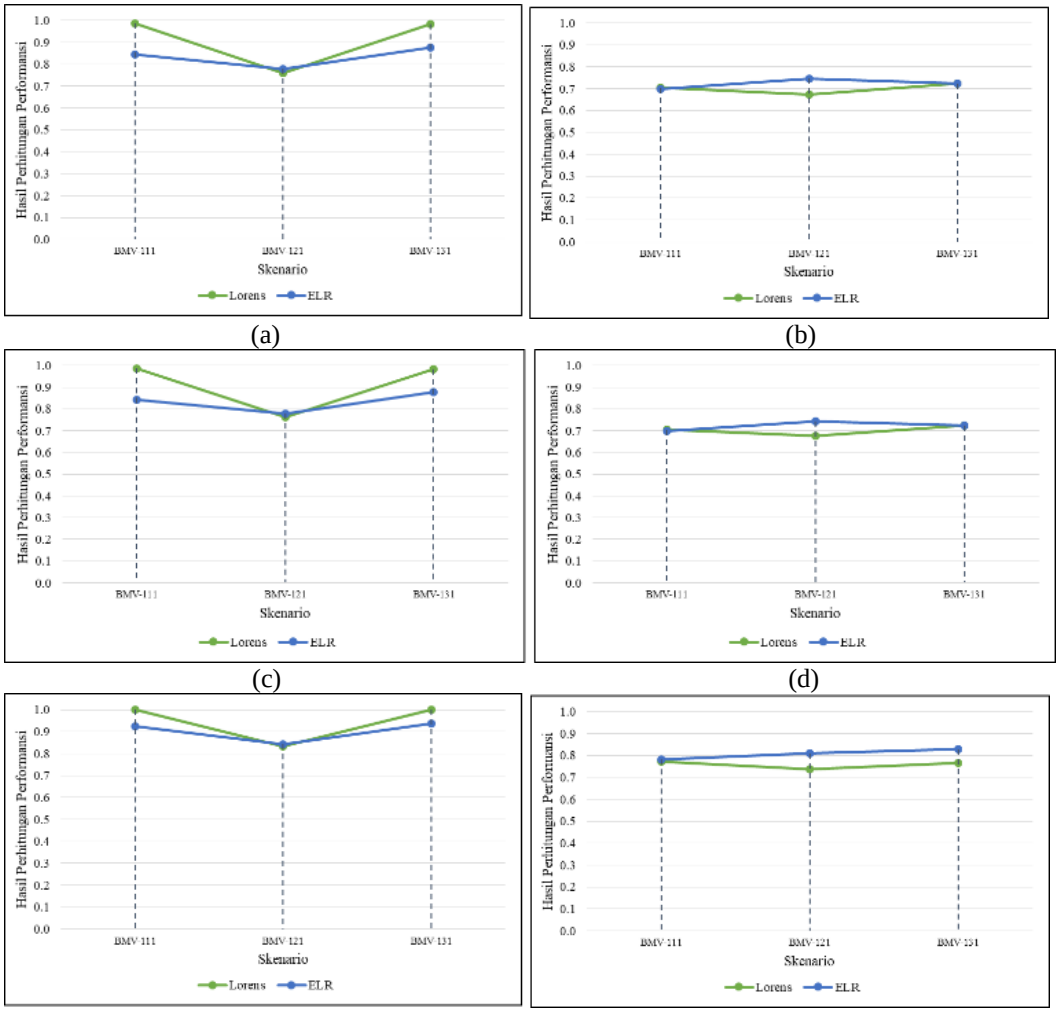
Pola perubahan besarnya nilai performansi model pada data dengan skenario BMV-121 dan BMV-131 ditunjukkan dalam Gambar 4.4, sedangkan skenario BMV-111 ditunjukkan dengan visualisasi yang sama dengan Gambar 4.1. Visualisasi hasil analisis pada Gambar 4.1 untuk skenario BMV-111 dan Gambar 4.4 untuk skenario BMV-121 dan BMV-131 menunjukkan pola yang cukup menarik. Analisis data *training* pada skenario BMV-111 dan BMV-131 menghasilkan pola yang hampir sama, yaitu rata-rata nilai performansi yang dihasilkan mendekati sempurna. Hal ini diduga karena proses pembangkitan data pada skenario BMV-131 untuk variabel ke-3 hingga 160 menggunakan distribusi yang sama dengan skenario BMV-111. Walaupun terdapat dua variabel yang dirancang untuk berkombinasi linier dengan variabel yang lain, pada akhirnya hasil yang didapatkan tidak menunjukkan perbedaan yang cukup signifikan dengan skenario BMV-111. Pola hasil prediksi data *testing* pada kedua skenario juga tidak

memiliki pola tertentu, tetapi menunjukkan kecenderungan *overfitting*. Hal ini ditunjukkan oleh rentang nilai performansi data *testing* yang tidak sebaik pada data *training*. Walaupun demikian, rata-rata nilai yang ditunjukkan dalam Gambar 4.4b dan 4.4d tidak sampai menyentuh angka kurang dari 0,5, sehingga hasil yang didapatkan masih dapat dipertimbangkan.

Jika hasil yang didapatkan pada skenario BMV-111 dan BMV-131 hampir sama, maka tidak demikian halnya pada skenario BMV-121. Pada data *training*, pola penurunan performansi terlihat cukup signifikan. Sesuai dengan Gambar 4.4a, jika variabel dibagi dalam 5 partisi, maka ketiga kriteria performansi menunjukkan hasil yang sama dan sempurna. Akan tetapi performansi tersebut lambat laun akan menurun jika jumlah partisi ditambah. Pada skenario 70 partisi, nilai total akurasi dan BCR mendekati 0,7, sedangkan AUC mendekati 0,8. Jika dipandang dari besarnya hasil akhir ketiga performansi tersebut, maka Lorens dapat dianggap sebagai metode yang masih baik walaupun data memiliki sifat multikolinearitas yang cukup tinggi. Berbeda dengan hasil yang didapatkan pada data *training*, hasil yang didapatkan pada data *testing* pada skenario BMV-121 memiliki pola yang hampir sama dengan skenario BMV-111 dan BMV-131. Bertambahnya jumlah partisi variabel tidak mempengaruhi performansi Lorens untuk memprediksi data baru di luar data *training* yang digunakan. Hasil yang ditunjukkan juga cukup stabil di sekitar nilai 0,65—0,85, mengingat pola pada data *training* yang cenderung menurun. Dengan demikian dapat disimpulkan bahwa secara keseluruhan, ELR dapat menghasilkan performansi yang cukup baik pada jenis-jenis data yang memiliki sifat multikolinieritas, seperti normal multivariat dan kombinasi linier.

**Tabel 4.3** Hasil Perhitungan Nilai Performansi Model ELR pada Data Simulasi dengan Skenario untuk Mengetahui Efek Multikolinearitas Data

| Metode | Skenario | Data *Training* | | | Data *Testing* | | |
|---|---|---|---|---|---|---|---|
| | | Total Akurasi | BCR | AUC | Total Akurasi | BCR | AUC |
| Lorens | BMV-111 | 0,9875 | 0,9875 | 0,9994 | 0,7700 | 0,7700 | 0,8500 |
| | BMV-121 | 0,7613 | 0,7613 | 0,8349 | 0,7050 | 0,7050 | 0,7680 |
| | BMV-131 | 0,9750 | 0,9750 | 0,9984 | 0,7250 | 0,7250 | 0,7880 |
| ELR | BMV-111 | 0,8438 | 0,8438 | 0,9240 | 0,7000 | 0,7000 | 0,7800 |
| | BMV-121 | 0,7775 | 0,7775 | 0,8436 | 0,7450 | 0,7450 | 0,8100 |
| | BMV-131 | 0,8775 | 0,8775 | 0,9368 | 0,7250 | 0,7250 | 0,8280 |




**Gambar 4.5** Visualisasi Perbandingan Performansi untuk Mengetahui Efek Multikolinieritas Data: (a) dan (b) Total Akurasi pada Data *Training* dan *Testing*, (c) dan (d) BCR pada Data *Training* dan *Testing*, serta (e) dan (f) AUC pada Data *Training* dan *Testing*

Perbandingan hasil analisis metode Lorens dan ELR pada Gambar 4.5 menunjukkan pola yang hampir sama pada data *training*. Kedua metode menunjukkan hasil yang baik pada data yang tidak memiliki sifat multikolinieritas atau memiliki sifat multikolinieritas akibat faktor kombinasi linier. Akan tetapi, besarnya nilai ukuran performansi Lorens pada jenis data tersebut lebih baik daripada ELR. Namun, penurunan performansi yang dihasilkan pada skenario BMV-121 terlihat lebih signifikan daripada ELR. Sedangkan pada data *testing*, perbedaan performansi Lorens dan ELR juga hanya terlihat pada skenario kedua berdasarkan kriteria total akurasi dan BCR, dengan hasil bahwa ELR menghasilkan performansi yang lebih besar daripada Lorens. Pada kriteria AUC, perbedaan

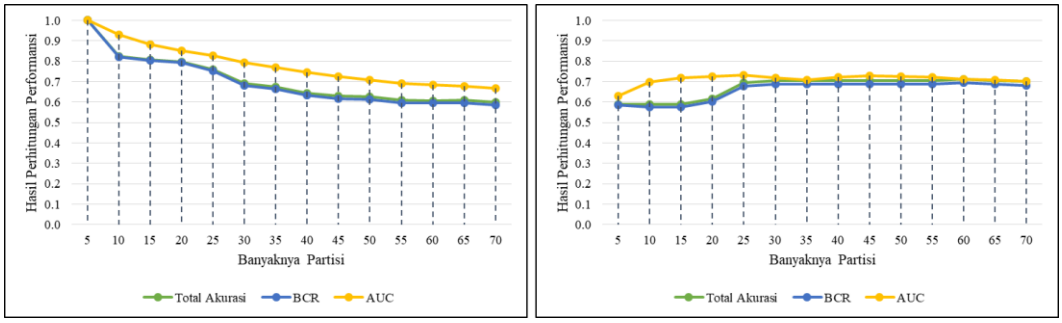
terlihat pada dua skenario, yaitu BMV-121 dan BMV-131 dengan ELR lebih baik daripada Lorens. Dengan demikian, pada pembahasan ini, ELR dapat dianggap sebagai metode yang lebih baik daripada Lorens.

## 4.3 Analisis Data Riil

Analisis data riil diterapkan pada permasalahan *drug discovery* yang berfokus pada sifat proteksi radiasi yang disebabkan oleh komponen senyawa terhadap sel. Dengan karakteristik yang mengacu pada dimensi tinggi, data ini digunakan sebagai objek untuk analisis dengan kedua metode *ensemble* berbasis regresi logistik, yaitu Lorens dan ELR. Pembahasan analisis secara lebih detail dijelaskan pada bagian Subbab 4.2.1 dan 4.2.2, serta pada Subbab 4.2.3 dijelaskan analisis perbandingan hasil kedua metode tersebut.

### 4.3.1 Metode Lorens

Proses analisis dengan menggunakan Lorens dilakukan dengan langkah yang sama dengan analisis simulasi, yaitu dengan diawali dengan penentuan skenario jumlah partisi. Proses perhitungan jumlah partisi minimum pada bagian ini sama persis dengan perhitungan yang dijelaskan pada Subbab 4.2.1 dan dihasilkan bahwa 5 partisi merupakan jumlah partisi minimum yang dapat digunakan pada data ini. Oleh karena itu, skenario 5, 10, ..., 70 partisi dapat juga digunakan pada data ini. Hasil analisis pada tiap-tiap partisi divisualisasikan dalam Gambar 4.6.




(a) (b)

**Gambar 4.6** (a) Efek Penambahan Partisi pada Data *Training* dan (b) Efek penambahan Partisi pada Data *Testing*

Pada Gambar 4.6a, dapat dilihat bahwa total akurasi dan BCR menghasilkan nilai yang hampir sama pada setiap analisisnya, sedangkan AUC selalu lebih tinggi daripada kedua ukuran performansi lainnya. Pada gambar tersebut, dapat dilihat pula kecenderungan efek penambahan jumlah partisinya. Diawali dari performansi yang sangat baik dan sempurna pada skenario 5 partisi, kemudian berangsur-angsur turun sesuai penambahan jumlah partisi tersebut hingga mencapai nilai di sekitar 0,6 pada skenario terakhir. Berbeda halnya dengan performansi yang ditampilkan dalam Gambar 4.6b, bertambahnya jumlah partisi terlihat memberikan efek yang baik pada data *testing*. Hal yang cukup menarik dalam visualisasi tersebut adalah bahwa performansi berangsur naik hingga skenario 25 partisi, kemudian perubahannya tidak terlalu signifikan pada skenario selanjutnya, yaitu cenderung mendekati nilai 0,7. Dengan demikian, hasil pengamatan ini dapat menunjukkan bahwa pada partisi lebih dari 25, model tidak cukup efektif untuk memberikan pengaruh yang berbeda pada data *testing*. Penjelasan lebih detail mengenai hasil rata-rata perhitungan performansi Lorens ditunjukkan dalam Tabel 4.4.

**Tabel 4.4** Rata-Rata Perhitungan Performansi Lorens pada Data Riil

| Partisi | *Training* | | | *Testing* | | |
|---|---|---|---|---|---|---|
| | Total Akurasi | BCR | AUC | Total Akurasi | BCR | AUC |
| 5 | **1,0000** | **1,0000** | **1,0000** | 0,5882 | 0,5861 | 0,6292 |
| 10 | 0,8239 | 0,8222 | 0,9312 | 0,5882 | 0,5764 | 0,6986 |
| 15 | 0,8060 | 0,8051 | 0,8806 | 0,5882 | 0,5764 | 0,7181 |
| 20 | 0,7955 | 0,7922 | 0,8530 | 0,6176 | 0,6042 | 0,7264 |
| 25 | 0,7582 | 0,7530 | 0,8265 | 0,6941 | 0,6764 | **0,7306** |
| 30 | 0,6910 | 0,6814 | 0,7939 | 0,7059 | 0,6875 | 0,7194 |
| 35 | 0,6746 | 0,6632 | 0,7696 | 0,7059 | 0,6875 | 0,7097 |
| 40 | 0,6433 | 0,6320 | 0,7441 | 0,7059 | 0,6875 | 0,7208 |
| 45 | 0,6284 | 0,6161 | 0,7243 | 0,7059 | 0,6875 | 0,7292 |
| 50 | 0,6254 | 0,6131 | 0,7091 | 0,7059 | 0,6875 | 0,7250 |
| 55 | 0,6104 | 0,5972 | 0,6925 | 0,7059 | 0,6875 | 0,7208 |
| 60 | 0,6075 | 0,5944 | 0,6832 | **0,7118** | **0,6938** | 0,7111 |
| 65 | 0,6090 | 0,5953 | 0,6772 | 0,7059 | 0,6875 | 0,7069 |
| 70 | 0,6000 | 0,5863 | 0,6687 | 0,7000 | 0,6813 | 0,7014 |

Nilai-nilai yang dicetak tebal pada Tabel 4.4 merupakan hasil perhitungan yang memiliki nilai paling tinggi pada semua skenario partisi. Pada data *training*, ketiga ukuran performansi memiliki nilai rata-rata paling tinggi pada analisis menggunakan 5 partisi, yaitu sebesar 1 atau 100%. Akan tetapi, ketika model yang didapatkan diterapkan ke dalam data *testing*, performansinya akan turun dengan sangat signifikan, yaitu mencapai angka di sekitar 0,6. Hal ini mengindikasikan terjadinya *overfitting* pada model yang didapatkan. Namun, apabila diperhatikan pada nilai terbesar dari data *testing*, didapatkan bahwa total akurasi dan BCR mencapai nilai tertinggi pada saat digunakan 60 partisi, sedangkan AUC mendapatkan nilai maksimum ketika digunakan 25 partisi.

**Tabel 4.5** Nilai Absolut Selisih Perbandingan Skenario

| Keterangan | Total Akurasi | BCR | AUC |
|---|---|---|---|
| Selisih skenario 25 dan 60 pada data *testing* | 0,0176 | 0,0174 | 0,0194 |
| Selisih skenario 5 dan 25 pada data *training* | 0,2418 | 0,2470 | 0,1735 |
| Selisih skenario 25 pada data *training* dan *testing* | 0,0641 | 0,0767 | 0,0960 |

Jika selisih nilai dari ketiga ukuran performansi dihitung pada kedua skenario tersebut, maka yang memiliki perubahan yang lebih besar adalah AUC. Hal ini ditunjukkan pada Tabel 4.5. Oleh karena itu, skenario terbaik ditentukan dengan berdasarkan nilai AUC tertinggi, yaitu pada partisi 25. Jika diperhatikan lebih lanjut pada partisi ini, selisih nilai ketiga ukuran performansi pada data *training* dan *testing* jauh lebih kecil jika dibandingkan dengan skenario 5 partisi, walaupun hasil yang didapatkannya jauh berbeda pada data *training*. Selain itu, jika dibandingkan dengan nilai ukuran performansi pada data *testing*, hasil perhitungannya cenderung lebih seimbang dan bahkan memiliki potensi yang lebih baik pada data *testing*. Oleh karena itu, dengan beberapa pertimbangan tersebut pemilihan skenario ini dapat dianggap sebagai skenario paling optimum pada data yang digunakan.

### 4.3.2 Metode ELR

Analisis data riil menggunakan metode ELR pada bagian ini dilakukan secara berulang sampai sebanyak 10 kali. Nilai probabilitas inisial yang digunakan adalah $1 - p\text{-}value$ dari perhitungan $t\text{-}test\ ranking$. Nilai tersebut digunakan

dengan harapan bahwa variabel-variabel yang memiliki perbedaan rata-rata yang signifikan berbeda pada kedua kategori memiliki peluang yang lebih besar untuk terpilih dalam algoritma ELR. Nilai probabilitas inisial yang didapatkan dari perhitungan *t-test ranking* untuk 10 data yang memiliki probabilitas paling besar ditunjukkan dalam Tabel 4.6.

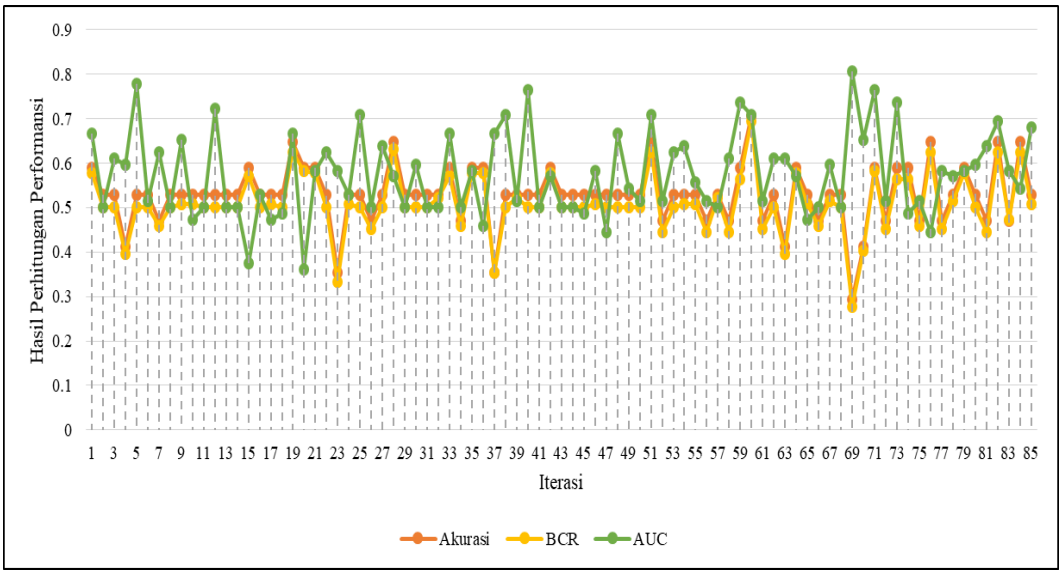
Hasil analisis menggunakan metode ELR didapatkan dari serangkaian proses pemodelan yang berulang. Proses ini melakukan pembaruan nilai probabilitas. Pada setiap iterasi yang dilakukan, ELR menghitung nilai performansi modelnya, khususnya BCR. Dalam penelitian ini, selain perhitungan BCR, peneliti juga melakukan perhitungan nilai total akurasi dan AUC pada setiap iterasinya. Visualisasi pola fluktuasi nilai ukuran performansi tersebut dicontohkan dalam Gambar 4.7, yaitu yang merupakan nilai performansi setiap iterasi pada pengulangan ke-3.

**Tabel 4.6** Nilai Inisial Probabilitas untuk Metode ELR

| No | Variabel | Probabilitas |
|---|---|---|
| 49 | ES_Sum_sCl | 0,9656 |
| 4 | Cl_Count | 0,9647 |
| 26 | ES_Count_sCl | 0,9647 |
| 196 | Energy | 0,9397 |
| 74 | SAscore | 0,9372 |
| 102 | Num_RingFusionBonds | 0,9353 |
| 125 | CHI_3_C | 0,9339 |
| 38 | ES_Sum_aaCH | 0,9187 |
| 161 | Dipole_mag | 0,9149 |
| 25 | ES_Count_sCH3 | 0,9020 |

Gambar 4.7 menunjukkan bahwa pada proses iterasi, nilai rata-rata BCR cenderung konstan di sekitar 0,46, sedangkan plot ukuran performansi model sangat fluktuatif nilainya, tetapi ketiga ukuran performansi tersebut memiliki nilai yang hampir sama. Jika dilihat secara keseluruhan, kebanyakan nilai-nilai total akurasi, BCR, dan AUC berada di atas 0,5. Bahkan pada satu iterasi tertentu, variabel yang terpilih menghasilkan model yang memiliki performansi mendekati 0,7. Akan tetapi, pada proses iterasi selanjutnya nilainya turun kembali. Hal ini menunjukkan

bahwa proses iterasi hingga mencapai angka rata-rata BCR yang konvergen dalam algoritma ELR tidak menjamin adanya suatu pola tertentu dari setiap ukuran performansi yang digunakan.




**Gambar 4.7** Visualisasi Fluktuasi Nilai Ukuran Performansi setiap Iterasi

Kelebihan ELR terletak pada kemampuannya yang dapat melakukan seleksi variabel. Oleh karena itu, dalam pembahasan ini ditunjukkan satu hasil seleksi variabel yang didapatkan dari metode ELR, yaitu seperti yang ditampilkan pada Tabel 4.7. Indeks yang dicetak tebal menunjukkan variabel yang memiliki probabilitas lebih dari 0,8 pada nilai probabilitas inisialnya.

**Tabel 4.7** Indeks Variabel Terpilih pada Analisis Pengulangan Ke-4

| Id Variabel Terpilih | | | | | | |
|---|---|---|---|---|---|---|
| 5 | **38** | 73 | 105 | 130 | **161** | 186 |
| 8 | 40 | 80 | 110 | 132 | 162 | 192 |
| **10** | 42 | 81 | 113 | 137 | 164 | 198 |
| **14** | **43** | 82 | 115 | 140 | 169 | 200 |
| **18** | **49** | 86 | 118 | 141 | **170** | 201 |
| **20** | 50 | 87 | 119 | 142 | 175 | 211 |
| 22 | 59 | 89 | 121 | 151 | **176** | 217 |
| **25** | 60 | 94 | 124 | **152** | 178 | |
| 30 | 65 | 96 | **125** | 157 | 182 | |
| 31 | 66 | 97 | 127 | 160 | 185 | |

Tabel 4.8 menunjukkan variabel-variabel yang cenderung terpilih pada satu rangkaian proses analisis ELR. Variabel yang dicetak tebal pada tabel tersebut menunjukkan variabel yang memiliki inisialisasi nilai probabilitas lebih dari 0,8. Berdasarkan Tabel 4.8, dapat dilihat bahwa dari 22 variabel dengan probabilitas tertinggi, sebanyak 20 variabel diantaranya termasuk ke dalam variabel yang memiliki probabilitas yang tinggi pada nilai inisialisasinya. Variabel-variabel tersebut dapat dianggap sebagai variabel yang cukup berarti pada proses analisis klasifikasi menggunakan metode ELR.

**Tabel 4.8** Variabel yang Konsisten Terpilih dalam Algoritma ELR

| No | Variabel | No | Variabel |
|---|---|---|---|
| 1 | **125** | 14 | **10** |
| 2 | **161** | 15 | **14** |
| 3 | **43** | 16 | **38** |
| 4 | **4** | 17 | **196** |
| 5 | **32** | 18 | 44 |
| 6 | **21** | 19 | 79 |
| 7 | **25** | 20 | **176** |
| 8 | **167** | 21 | **139** |
| 9 | **20** | 22 | **197** |
| 10 | **74** | ⋮ | ⋮ |
| 11 | **84** | 214 | 207 |
| 12 | **152** | 215 | 144 |

Setelah melakukan analisis secara berulang hingga 10 kali, dihitung nilai rata-rata analisis dari nilai performansi terakhir pada iterasi ke-1 sampai 10, Hasil perhitungan yang didapatkan, ditunjukkan dalam Tabel 4.9. Berdasarkan hasil perhitungan tersebut, didapatkan bahwa performansi model cenderung lebih baik pada data *training* daripada data *testing*. Ini mengindikasikan adanya *overfitting* pada model yang didapatkan.

**Tabel 4.9** Rata-Rata Hasil Perhitungan Performansi Metode ELR

| Kriteria | *Training* | *Testing* |
|---|---|---|
| Total Akurasi | 0,58507 | 0,52941 |
| BCR | 0,55609 | 0,50417 |
| AUC | 0,66470 | 0,63056 |

### 4.3.3 Perbandingan Hasil Analisis Data Riil

Perbandingan hasil analisis Lorens dan ELR pada bagian ini digunakan untuk mengetahui metode yang lebih baik jika diaplikasikan pada data riil berdasarkan nilai ukuran performansinya. Hasil analisis Lorens yang digunakan merupakan pada perbandingan ini adalah hasil terbaik yang telah didapatkan pada pembahasan dalam Subbab 4.3.1, yaitu Lorens dengan 25 partisi, sedangkan hasil ELR yang digunakan adalah nilai rata-rata perhitungan performansi dari 10 kali pengulangan analisis. Tabel perbandingan kedua metode ditampilkan pada Tabel 4.10.

**Tabel 4.10** Perbandingan Lorens dan ELR pada Data Riil

| Kriteria | Lorens | | ELR | |
|---|---|---|---|---|
| | *Training* | *Testing* | *Training* | *Testing* |
| Total Akurasi | 0,7582 | 0,6941 | 0,5851 | 0,5294 |
| BCR | 0,7530 | 0,6764 | 0,5561 | 0,5042 |
| AUC | 0,8265 | 0,7306 | 0,6647 | 0,6306 |

Perbandingan hasil analisis pada Tabel 4.10 menunjukkan metode Lorens memiliki performansi yang lebih tinggi, baik pada data *training* maupun data *testing*. Nilai AUC yang didapatkan dari Lorens mencapai angka tertinggi pada data *training*, yaitu sebesar 0,8265. Begitu pula pada data *testing*, AUC Lorens menghasilkan nilai paling tinggi diantara performansi yang lain ataupun dibandingkan dengan performansi yang didapatkan dari metode ELR. Dengan demikian, dalam analisis pada data riil, khususnya pada data yang digunakan dalam penelitian ini, dapat disimpulkan bahwa Lorens merupakan metode yang lebih baik daripada ELR.

*(Halaman ini sengaja dikosongkan)*

# BAB 5
# KESIMPULAN DAN SARAN

## 5.1 Kesimpulan

Analisis klasifikasi dilakukan pada data riil dari permasalahan *drug discovery* mengenai efek senyawa terhadap proteksi radiasi sel dan juga dilakukan pada data simulasi. Pada hasil analisis simulasi mengenai efek penambahan jumlah variabel, ELR merupakan metode yang lebih baik dibandingkan dengan ELR karena menghasilkan performansi yang lebih stabil. Pada simulasi mengenai efek keseimbangan data, ELR lebih unggul daripada Lorens karena penurunan yang terjadi pada rasio 1:2 tidak terlalu signifikan. Sedangkan pada simulasi mengenai efek multikolinieritas, ELR juga lebih baik daripada Lorens karena kecenderungan *overfitting* lebih kecil. Sedangkan pada data riil, hasil analisis yang didapatkan menunjukkan bahwa dengan kriteria total akurasi, BCR, dan AUC, Lorens memiliki performansi lebih baik daripada ELR.

## 5.2 Saran

Pada algoritma ELR, kendala yang mungkin muncul dalam proses perhitungan update probabilitas adalah apabila nilai quality bertanda negatif, dan estimasi parameter $\beta_{j,l}$ yang dihasilkan kurang dari nilai *quality*, maka $prob_{j,l} + quality.\beta_j^{2.sign(quality)}$ akan cenderung menjadi besar karena unsur $\beta_j^{2.sign(quality)}$ akan berubah menjadi pembagi untuk nilai *quality*. Nilai hasil perhitungan $quality.\beta_j^{2.sign(quality)}$ yang cenderung membesar akan sangat mendominasi nilai $prob_{j,l-1}$, sehingga perubahan nilai probabilitas bisa menjadi sangat berbeda dengan probabilitas sebelumnya. Oleh karena itu, perlu dilakukan kajian ulang untuk langkah update probabilitas pada metode ELR agar perubahan nilai probabilitas variabel tidak didominasi oleh nilai $quality.\beta_j^{2.sign(quality)}$.

Pada penelitian selanjutnya perlu dilakukan studi simulasi dengan data sampel yang lebih besar dari 100, sehingga dapat menjadi gambaran pada aplikasinya dalam big data. Selain itu, perlu dilakukan studi simulasi degan

skenario yang lebih bervariasi untuk mengetahui kelebihan dan kekurangan kedua metode, Lorens dan ELR, dengan kasus data yang lebih kompleks. Skenario berdasarkan data riil yang digunakan juga disarankan untuk dilakukan, agar dapat mengetahui bagaimana sikronisasi hasil antara studi simulasi dengan analisis data riil. Pada analisis dengan ELR, perlu diperlihatkan bagaimana sebaran nilai beta, karena efek regularisasi $l_2$ terletak pada perubahan nilai *standard error* parameter.

# DAFTAR PUSTAKA


Ahn, H., Moon, H., Fazzari, M. J., Lim, N., Chen, J. J., dan Kodell, R. L. (2007), "Classification by *Ensemble*s from Random Partitions of High-Dimensional Data", *Computational Statistics and Data Analysis,* Vol. 51, No. 12, hal. 6166—6179.

Alvarsson, J., Lampa, S., Schaal, W., Andersson, C., Wikberg, J. E. S., dan Spjuth, O. (2016), "Large-Scale Ligand-Based Predictive Modelling using Support Vector Machine", *Journal of Cheminformatics,* Vol. 8, No. 1, hal. 39.

Andrew, G. dan Gao, J. (2007), "Scalable *Training* of $L_1$-Regularized Log-Linear Models", *Proceedings of The 24$^{th}$ International Conference on Machine Learning*, hal 33—40.

Ariyasu, S., Sawa, A., Morita, A., Hanaya, K., Hoshi, M., Takahashi, I., Wang, B., dan Aoki, S. (2014), "Design and Synthesis of 8-Hydroxyquinoline-Based Radioprotective Agents", *Bioorganic and Medicinal Chemistry,* Vol. 22, No. 15, hal. 3891—3905.

Bai, L. Y., Dai, H., Xu, Q., Junaid, M., Peng, S. L., Zhu, X., Xiong, Y., dan Wei, D. Q. (2018), "Prediction of Effective *Drug* Combinations by Improved Naïve Bayesian Algorithm", *International Journal of Molecular Sciences,* Vol. 19, No. 2, hal. 467.

Bielza, C., Robles, V., dan Larrañaga, P. (2011), "Regularized Logistic Regression Without a Penalty Term: An Application to Cancer Classification With Microarray Data", *Expert Systems with Applications,* Vol. 38, No.5, hal. 5110—5118.

Bühlmann, P. (2012), "Bagging, Boosting, and *Ensemble* Methods", *Handbook of Computational Statistics,* Springer: Berlin, hal. 985—1022.

Burbidge, R., Trotter, M., Buxton, B., dan Holden, S. (2001), "*Drug* Design by Machine Learning: Support Vector Machines for Pharmaceutical Data Analysis", Computer and Chemistry, Vol. 26, No. 1, hal. 5—14.

Cheng, F. dan Sutariya, V. (2012), "Applications of Artificial Neural Network Modeling in *Drug Discovery*", *Clin. Exp. Pharmacol,* Vol. 2, No. 3, hal. 1—2.

Dietterich, T. G. (2000), "*Ensemble* Methods in Machine Learning", *International Workshop on Multiple Classifier Systems,* hal. 1-15.

Duan, Q., Anjami, N. K., Gao, X., dan Sorooshian, S. (2007), "Multi-Model *Ensemble* Hydrologic Prediction Using Bayesian Model Averaging", *Advances in Water Resources,* Vol. 30, No. 5, hal. 1371—1386.

Fan, Q., Wang, Z., Li, D., Gao, D., dan Zha, H. (2016), "Entropy-based Fuzzy Support Vector Mashine for Imbalanced Datasets", *Knowledge-Based Systems*, Vol. 115, hal. 87—89.

Feriante, J. (2015), *Massively Multitask Deep Learning for Drug Discovery*, Tesis Master, University of Wisconsin-Madison, Wisconsin.

Gmuender, H. (2002), "Perspectives and Challenges for DNA Microarrays in *Drug Discovery* and Development", Biotechniques, Vol. 32, No. 1, hal. 152—159.

Härdle, W. K., Prastyo, D. D., dan Hafner, C. M. (2013), "Support Vector Machines with Evolutionary Model Selection for Default Prediction", *The Oxford Handbook of Applied Nonparametric and Semiparametric Econometrics and Statistics.* Oxford University Press Inc. New York, NY.

Härdle, W. K. dan Prastyo, D. D. (2014), "Embedded Predictor Selection for Default Risk Calculation: A Southeast Asian Industry Study", *Handbook of Asian Finance*, Vol. 1, hal 131—148.

Hauck, W. W., Jr. dan Donner, A. (1977), "Wald's Test as Applied to Hypotheses in Logit Analysis", *Journal of The American Statistical Association*, Vol. 72, No. 360a, hal. 851—853.

Hosmer, D. W. dan Lemeshow, S. (2000), *Applied Logistic Regression*, 2nd edition, John Wiley and Sons, New York.

Huynh, T., He, Y., Willis, A., dan Rüger, S. (2016), "Adverse *Drug* Reaction Classification with Deep Neural Network", *Proceedings of COLING 2016, The 26th International Conference on Computational Linguistics: Technical Papers*, hal. 877—887.

Jayaraj, P. B., Ajay, M. K., Nufail, M., Gopakumar, G., dan Jallel, U. C. A. (2016), "GPURFSCREEN: A GPU Based Virtual Screening Tool Using Random Forest Classifier", *Journal of Cheminformatics*, Vol. 8, No. 1, hal. 12.

Johnson, R. A. dan Wichern, D. W. (2007), *Applied Multivariate Statistical Analysis*, 6th edition, Prentice-Hall, New Jersey.

Kannanthanathu, A. F. (2017), *Wavelet Transform and Ensemble Logistic Regression for Driver Drowsiness Detection*, Tesis Master, Rashtrasanth Tukadoji Maharaj Nagpur University, Maharashtra.

Kimura, M., Aoki, S., dan Ohwada, H. (2017), "Predicting Radiation Protection and Toxicity of p53 Targeting Radioprotectors Using Machine Learning",

*Computational Intelligence in Bioinformatics and Computational Biology (CIBCB),* hal. 1—6.

Koh, K., Kim, S. J., dan Boyd, S. (2007), "An Interior-Point Method for Large-Scale $\ell_1$-Regularized Logistic Regression", *Journal of Machine Learning Research,* No. 8, hal. 1519—1555.

Kuswanto, H., Asfihani, A., Sarumaha, Yogi., dan Ohwada, H. (2015), "Logistic Regression *Ensemble* for Predicting Customer Defection with Very Large Sample Size", *Procedia Computer Science,* Vol. 72, hal. 86—93.

Kuswanto, H. dan Werdhana, R. W. (2017), "Logistic Regression *Ensemble* to Classifiy Alzheimer Gene Expression", *International Conference on Smart Cities, Automation, and Intelligent Computing Systems (ICON-SONICS),* hal. 36—41.

Kuswanto, H., Melasasi, J. N., Ohwada, H. (2018), "Enzyme Classification on DUD-E Database Using Logistic Regression *Ensemble* (Lorens)", *In Innovative Computing, Optimization, and, Its Applications,* hal. 93—109.

Laarhoven, T. dan Marchiori, E. (2013), "Predicting *Drug*-Target Interactions for New *Drug* Compounds Using A Weighted Nearest Neighbor Profile", *PloS One,* Vol. 8, No. 6, hal. e66951.

Lee, S. I., Lee, H., Abbeel, P., dan Ng, A. Y. (2006), "Efficient $L_1$ Regularized Logistic Regression", *American Association for Artificial Intelligence,* Vol. 6, hal. 401—408.

Lim, N. (2007), *Classification by Ensemble from Random Partitions Using Logistic Regression Models,* Disertasi Ph.D., Stony Brook University, New York.

Lim, N., Ahn, H., Moon, H., dan Chen, J. J. (2009), "Classification of High-Dimensional Data with *Ensemble* of Logistic Regression Models", *Journal of Biopharmaceutical Statistics,* Vol. 20, No. 1, hal. 160—171.

Lin, W. J. dan Chen, J. J. (2011), "Class-Imbalanced Classifiers for High Dimensional Data", *Briefings in Bioinformatics,* Vol. 14, No. 1, hal. 13—26.

Matsumoto, A., Ito, T., Nishi, Y., Teraoka, T., Aoki, S., dan Ohwada, H. (2015), "Prediction of Radioprotectors Targeting p53 for Suppression of Acute Effect of Cancer Radiotherapy using Machine Learning", *Bioinformatics and Biomedicine (BIBM), 2015 IEEE International Conference on,* hal. 1725—1727.

Matsumoto, A., Aoki, S., dan Ohwada, H. (2016), "Comparison of Random Forest and SVM for Raw Data in *Drug Discovery*: Prediction of Radiation Protection and

Toxicity Case Study", *International Journal of Machine Learning and Computing*, Vol. 6, No. 2, hal. 145.

Montgomery, D. C., Peck, E. A., dan Vining, G. G. (2012), *Introduction to Linear Regression Analysis*, 5th edition, John Wiley and Sons, New Jersey.

Ng, A. Y. (2004), "Feature Selection, $L_1$ vs. $L_2$ Regularization and Rotational Invariance", *Proceedings of The Twenty-First International Conference on Mahine Learning*, hal.78.

Rokach, L. (2010), "*Ensemble*-Based Classifiers", *Artificial Intelligence Review*, Vol. 33, No. 1—2, hal. 1—39.

Romero, C., Ventura, S., Pechenizky, M., dan Baker, Ryan. (2011), *Handbook of Educational Data Mining*, Boca Raton: CRC Press.

Scotti, L., Ishiki, H., Mendonça, J. B. Junior, Silva, M. S., dan Scotti, M. T. (2015), "Artificial Neural Network Methods Applied to *Drug Discovery* for Neglected Diseases", Combinatorial Chemistry and High Throughput Screening, Vol. 18, No. 8, hal. 819—829.

Shu, C. dan Burn, D. H. (2004), "Artifial Neural Network *Ensemble*s and Their Application in Pooled Flood Frequency Analysis", *Water Resources Research*, Vol. 40, No.9.

Suhartono, Faulina, R., Lusia, D. A., Otok, B. W., Sutikno., dan Kuswanto, H, (2012), "*Ensemble* Method Based on ANFIS-ARIMA for Rainfall Prediction", *In Statistics in Science, Business, and Engineering (ICSSBE), 2012 International Conference on IEEE*, hal. 1—4.

Tanwani, A. K. dan Farooq, M. (2009), "The Role of Biomedical Dataset in Classification", *Conference on Artificial Intelligence in Medicine in Europe*, hal. 370—374.

Tibshirani, R. (1996), "Regression Shrinkage and Selection via The Lasso", *Journal of The Royal Statistical Societ. Series B (Methodological)*, hal. 267—288.

Warmuth, M. K., Rätsch, G., Mathieson, M., Liao, J., dan Lemmen, C. (2003), "Active Learning in The *Drug Discovery* Process", In Advances in Neural Information Processing Systems, hal. 1449—1456.

Witten, I. H., Frank, E., Mark, A. H., dan Pal, C. J. (2016), *Data Mining: Practical Machine Learning Tools and Techniques*, Morgan Kaufmann.

Zakharov, R. dan Dupont, P. (2011), "*Ensemble* Logistic Regression for Feature Selection", *Pattern Recognition in Bioinformatics,* No. 7036, hal. 133—144.

*(Halaman ini sengaja dikosongkan)*

# LAMPIRAN

**Lampiran 1** Ilustrasi Analisis Data dengan Menggunakan Metode Lorens

Ilustrasi analisis data dengan menggunakan Lorens ditunjukkan pada uraian di bawah ini. Langkah yang dijelaskan berikut ini disesuaikan dengan algoritma yang telah ditunjukkan dalam Tabel 1. Secara rinci, proses perhitungan pada analisis dengan menggunakan metode Lorens dijelaskan dalam uraian berikut ini.

**Tabel 1** Data Sampel sebagai Contoh Ilustrasi Proses Analisis

| No | $y$ | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ | $x_7$ | $x_8$ | $x_9$ | ... | $x_{15}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0,87 | 0,58 | 0,14 | 0,97 | 0,22 | 0,49 | 0,72 | 0,04 | 0,07 | ... | 0,69 |
| 2 | 1 | 0,21 | 0,71 | 0,73 | 0,43 | 0,48 | 0,02 | 0,09 | 0,77 | 0,14 | ... | 0,00 |
| 3 | 1 | 0,46 | 0,64 | 0,31 | 0,64 | 0,60 | 0,49 | 0,91 | 0,73 | 0,09 | ... | 0,17 |
| 4 | 1 | 0,01 | 0,28 | 0,57 | 0,81 | 0,03 | 0,39 | 0,29 | 0,6 | 0,04 | ... | 0,28 |
| 5 | 1 | 0,55 | 0,61 | 0,89 | 0,28 | 0,27 | 0,45 | 0,14 | 0,82 | 0,41 | ... | 0,29 |
| 6 | 0 | 0,99 | 0,68 | 0,06 | 0,44 | 0,11 | 0,19 | 0,28 | 0,32 | 0,35 | ... | 0,10 |
| 7 | 0 | 0,45 | 0,08 | 0,88 | 0,12 | 0,94 | 0,51 | 0,38 | 0,23 | 0,82 | ... | 0,67 |
| 8 | 0 | 0,73 | 0,04 | 0,00 | 0,95 | 0,76 | 0,85 | 0,71 | 0,58 | 0,97 | ... | 0,32 |
| 9 | 0 | 0,36 | 0,69 | 0,26 | 0,32 | 0,16 | 0,31 | 0,68 | 0,99 | 0,08 | ... | 0,17 |
| 10 | 0 | 0,73 | 0,96 | 0,52 | 0,81 | 0,67 | 0,64 | 0,11 | 0,24 | 0,5 | ... | 0,48 |

*Input* 1: Data sampel yang digunakan merupakan data contoh yang dibangkitkan secara random dengan jumlah data sebanyak 10 dan jumlah variabel sebanyak 15. Notasi $y$ dalam Tabel 1 menunjukkan target atau variabel respon, sedangkan $x_1, x_2, \dots, x_{15}$ merupakan variabel prediktor. Data contoh yang terdapat dalam Tabel 1 dapat dikategorikan kedalam data berdimensi tinggi karena jumlah variabel lebih banyak daripada jumlah data atau dapat dituliskan sebagai $p > n$.

*Input* 2: Jumlah *subspace* yang ditentukan pada ilustrasi ini adalah sebanyak 5. Jumlah ini dipertimbangkan dari perhitungan jumlah minimal *subspace* yang bisa digunakan pada data ilustrasi yang digunakan. Sesuai Tabel 1, diketahui bahwa jumlah data yang akan dianalisis adalah $n = 10$ dan $p = 15$. Jika semua data digunakan sebagai data *training*, maka jumlah variabel maksimal yang terdapat dalam satu *subspace* adalah sebanyak $n - 1 = 10 - 1 = 9$, karena jumlah variabel dalam satu *subspace* tidak boleh sama dengan atau lebih dari jumlah data. Dengan

demikian, jumlah *subspace* minimal yang bisa digunakan agar memenuhi syarat $p < n$ adalah $m = \frac{15}{9} = 1.68 \approx 2$. Jumlah maksimal *subspace* yang bisa digunakan adalah $m = p = 15$, yaitu yang akan menghasilkan 1 variabel pada setiap *subspace*. Dengan demikian, penentuan 5 *subspace* dalam ilustrasi ini dapat digunakan karena berada diantara rentang 2 hingga 15.

*Input* 3: Menentukan nilai *threshold* berdasarkan Persamaan (2.12), seperti langkah di bawah ini.

$$threshold = \frac{\frac{1}{10}\sum_{i=1}^{10} y_i + 0{,}5}{2} = \frac{\frac{5}{10} + 0{,}5}{2} = 0{,}5$$

Dengan demikian nilai threshold yang digunakan untuk menentukan prediksi target 1 dan 0 adalah 0,5, dengan $\hat{y}_i = 1$ apabila $\hat{P}(Y_i|x_i) \geq 0{,}5$ dan sebaliknya untuk $\hat{y}_i = 0$ untuk $i = 1,2,\ldots,10$.

Dengan menggunakan ketiga input yang telah dijelaskan di atas, maka proses analisis untuk *ensemble* pertama secara rinci dilakukan seperti pada langkah berikut ini.

Langkah 1: Membagi data menjadi 5 *subspace* secara random dan *mutually exclusive* dengan jumlah anggota *subspace* yang sama, seperti yang ditunjukkan pada Tabel 2. Sesuai dengan Tabel 2, jumlah variabel yang masuk dalam *subspace* 1 ada sebanyak 3 variabel, yaitu $x_1, x_7,$ dan $x_{14}$, *subspace* 2 ada sebanyak 3 variabel, yaitu $x_5, x_8,$ dan $x_{12}$, *subspace* 3 ada sebanyak 3 variabel, yaitu $x_2, x_6,$ dan $x_{11}$, *subspace* 4 ada sebanyak 3, yaitu $x_3, x_9,$ dan $x_{13}$, sedangkan *subspace* 5 ada sebanyak 3, yaitu $x_4, x_{10},$ dan $x_{15}$.

**Tabel 2** Pembagian Data ke dalam 3 *Subspace*

| Variabel | *Subspace* | Variabel | *Subspace* |
|---|---|---|---|
| $x_1$ | 1 | $x_9$ | 4 |
| $x_2$ | 3 | $x_{10}$ | 5 |
| $x_3$ | 4 | $x_{11}$ | 3 |
| $x_4$ | 5 | $x_{12}$ | 2 |
| $x_5$ | 2 | $x_{13}$ | 4 |
| $x_6$ | 3 | $x_{14}$ | 1 |
| $x_7$ | 1 | $x_{15}$ | 5 |
| $x_8$ | 2 | | |

Langkah 2: Memodelkan data pada tiap-tiap *subspace* yang telah terbentuk. Dalam ilustrasi ini dicontohkan hasil pemodelan untuk *subspace* yang pertama, dengan anggota $x_1, x_7,$ dan $x_{14}$. Tabel estimasi parameter dan pengujian individu pada analisis ini ditunjukkan dalam Tabel 3.

**Tabel 3** Estimasi Parameter dan Pengujian Individu Hasil Analisis dengan Lorens

| Parameter | Estimasi | *Standard Error* | *z* | *p-value* |
|---|---|---|---|---|
| $\beta_0$ | 2,8801 | 2,9896 | 0,9630 | 0,3350 |
| $\beta_1$ | -3,1043 | 2,8262 | -1,0980 | 0,2720 |
| $\beta_7$ | 0,9389 | 1,3422 | 0,6990 | 0,4840 |
| $\beta_{14}$ | -6,4705 | 6,8640 | -0,9430 | 0,3460 |

Berdasarkan Tabel 3, dapat diketahui bahwa pada *subspace* pertama, variabel $x_1, x_7,$ dan $x_{14}$ tidak berpengaruh secara signifikan terhadap variabel respon atau variabel $y$. Hal ini dapat dilihat dari $p$-*value* yang seluruhnya lebih dari tingkat signifikansi yang ditentukan, yaitu sebesar 5%. Model probabilitas regresi logistik berdasarkan Tabel 3 dapat dituliskan sebagai berikut.

$$\hat{P}(Y_i = 1|\boldsymbol{x}_i) = \frac{1}{1 + e^{-(2{,}8801 - 3{,}1043x_1 + 0{,}9389x_7 - 6{,}4705x_{14})}}$$

$$\hat{P}(Y_i = 0|\boldsymbol{x}_i) = \frac{e^{-(2{,}8801 - 3{,}1043x_1 + 0{,}9389x_7 - 6{,}4705x_{14})}}{1 + e^{-(2{,}8801 - 3{,}1043x_1 + 0{,}9389x_7 - 6{,}4705x_{14})}}$$

Langkah 3: Mendapatkan nilai prediksi berdasarkan model yang telah didapatkan pada Langkah 2. Perhitungan nilai probabilitas dapat dilakukan seperti langkah berikut ini.

$$\hat{P}(Y_1 = 1|\boldsymbol{x}_1) = \frac{1}{1 + e^{-(2{,}8801 - 3{,}1043(1{,}10636) + 0{,}9389(0{,}96807) - 6{,}4705(0{,}1))}}$$
$$= 0{,}42739$$

$$\hat{P}(Y_2 = 1|\boldsymbol{x}_2) = \frac{1}{1 + e^{-(2{,}8801 - 3{,}1043(-1{,}07986) + 0{,}9389(-1{,}14226) - 6{,}4705(0{,}72))}}$$
$$= 0{,}62272$$

$$\vdots$$

$$\hat{P}(Y_{10} = 1|\boldsymbol{x}_{10}) = \frac{1}{1 + e^{-(2{,}8801 - 3{,}1043(0{,}64262) + 0{,}9389(-1{,}07526) - 6{,}4705(0{,}34))}}$$
$$= 0{,}08913$$

Hasil perhitungan secara keseluruhan untuk setiap data dan *subspace* ditunjukkan dalam Tabel 4.

**Tabel 4** Prediksi Nilai Probabilitas pada Setiap *Subspace*

| No | *Subspace 1* | *Subspace 2* | *Subspace 3* | *Subspace 4* | *Subspace 5* |
|---|---|---|---|---|---|
| 1 | 0,4274 | 0,5936 | 0,3181 | 0,5660 | 0,3754 |
| 2 | 0,6227 | 0,8964 | 0,9118 | 0,9948 | 0,9065 |
| 3 | 0,4336 | 0,7564 | 0,2374 | 0,7259 | 0,3582 |
| 4 | 0,8453 | 0,9911 | 0,7952 | 0,9940 | 0,7591 |
| 5 | 0,8527 | 0,2518 | 0,8854 | 0,9195 | 0,6775 |
| 6 | 0,0011 | 0,5994 | 0,3430 | 0,0034 | 0,2823 |
| 7 | 0,5754 | 0,0109 | 0,3823 | 0,0148 | 0,2055 |
| 8 | 0,4877 | 0,4239 | 0,1441 | 0,0000 | 0,6972 |
| 9 | 0,6648 | 0,4509 | 0,5169 | 0,7503 | 0,1367 |
| 10 | 0,0891 | 0,0255 | 0,4659 | 0,0314 | 0,6016 |

Langkah 4: Menghitung nilai rata-rata probabilitas untuk setiap data. Berdasarkan Tabel 4, perhitungan nilai rata-rata probabilitas dapat dituliskan seperti langkah berikut ini.

Data ke-1: $\frac{1}{5}(0{,}4274 + 0{,}5936 + \cdots + 0{,}3754) = 0{,}4561$

Data ke-2: $\frac{1}{5}(0{,}6227 + 0{,}8964 + \cdots + 0{,}9065) = 0{,}8664$

$\vdots$

Data ke-10: $\frac{1}{5}(0{,}0891 + 0{,}0255 + \cdots + 0{,}6016) = 0{,}2427$

Hasil perhitungan secara keseluruhan dituliskan dalam Tabel 5.

**Tabel 5** Rata-Rata Nilai Prediksi Probabilitas

| No | Rata-Rata Probabilitas | No | Rata-Rata Probabilitas |
|---|---|---|---|
| 1 | 0,4561 | 6 | 0,2458 |
| 2 | 0,8664 | 7 | 0,2378 |
| 3 | 0,5023 | 8 | 0,3506 |
| 4 | 0,8769 | 9 | 0,5039 |
| 5 | 0,7174 | 10 | 0,2427 |

Langkah 5: Menentukan prediksi kategori berdasarkan nilai *threshold* yang telah ditentukan. Pada bagian Input 3, nilai *threshold* yang ditentukan adalah

sebesar 0,5. Dengan demikian, prediksi kategori yang didapatkan berdasarkan Tabel 5 adalah sebagai berikut.

**Tabel 6** Rata-Rata Nilai Prediksi Probabilitas

| No | Perbandingan terhadap Nilai *Threshold* | Prediksi Kategori |
|---|---|---|
| 1 | $< 0{,}5$ | 0 |
| 2 | $> 0{,}5$ | 1 |
| 3 | $> 0{,}5$ | 1 |
| 4 | $> 0{,}5$ | 1 |
| 5 | $> 0{,}5$ | 1 |
| 6 | $< 0{,}5$ | 0 |
| 7 | $< 0{,}5$ | 0 |
| 8 | $< 0{,}5$ | 0 |
| 9 | $> 0{,}5$ | 1 |
| 10 | $< 0{,}5$ | 0 |

Hasil yang didapatkan pada Tabel 6 merupakan hasil akhir untuk satu rangkaian proses *ensemble* pertama. Pada analisis ini, proses tersebut diulangi hingga 11 kali. Jumlah pengulangan ini ditentukan dengan pertimbangan agar pada saat proses *majority voting* tidak ada kemungkinan bahwa kedua kategori memiliki jumlah *vote* yang sama. Selanjutnya untuk menentukan prediksi Lorens paling akhir, digunakan gabungan dari hasil kesebelas *ensemble* yang didapatkan dengan cara *majority voting*. Langkah analisis untuk *ensemble* ke-1, 2, …, 11 adalah sebagai berikut.

Langkah 6: Melakukan Langkah 1 sampai 5 hingga didapatkan prediksi kategori untuk 5 *ensemble*. Pembagian data kedalam *subspace* pada masing-masing pengulangan ditunjukkan pada Tabel 7.

**Tabel 7** Pembagian *Subspace* dengan Sebelas Pengulangan

| Variabel | Pengulangan Ke- | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| $x_1$ | 1 | 5 | 2 | 3 | 1 | 5 | 2 | 3 | 4 | 4 | 2 |
| $x_2$ | 3 | 4 | 2 | 3 | 2 | 1 | 1 | 1 | 5 | 3 | 5 |
| $x_3$ | 4 | 2 | 4 | 1 | 1 | 1 | 4 | 3 | 3 | 3 | 5 |
| $x_4$ | 5 | 3 | 3 | 2 | 1 | 4 | 3 | 4 | 2 | 1 | 4 |
| $x_5$ | 2 | 2 | 3 | 2 | 5 | 4 | 2 | 5 | 3 | 5 | 3 |
| $x_6$ | 3 | 3 | 1 | 5 | 2 | 1 | 2 | 3 | 1 | 4 | 1 |

**Lanjutan Tabel 7** Pembagian *Subspace* dengan Sebelas Pengulangan

| Variabel | Pengulangan Ke- | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| $x_7$ | 1 | 1 | 5 | 4 | 4 | 4 | 5 | 5 | 5 | 2 | 5 |
| $x_8$ | 2 | 3 | 3 | 4 | 3 | 3 | 3 | 4 | 2 | 1 | 4 |
| $x_9$ | 4 | 1 | 2 | 1 | 2 | 2 | 5 | 4 | 4 | 2 | 3 |
| $x_{10}$ | 5 | 5 | 4 | 4 | 3 | 5 | 4 | 2 | 1 | 4 | 2 |
| $x_{11}$ | 3 | 5 | 1 | 1 | 5 | 3 | 3 | 5 | 2 | 5 | 4 |
| $x_{12}$ | 2 | 2 | 5 | 5 | 3 | 5 | 1 | 1 | 4 | 3 | 3 |
| $x_{13}$ | 4 | 4 | 1 | 5 | 4 | 2 | 4 | 2 | 3 | 5 | 1 |
| $x_{14}$ | 1 | 1 | 4 | 3 | 5 | 3 | 5 | 2 | 5 | 2 | 1 |
| $x_{15}$ | 5 | 4 | 5 | 2 | 4 | 2 | 1 | 1 | 1 | 1 | 2 |

Berdasarkan pembagian *subspace* tersebut, dengan langkah yang sama dengan penjelasan sebelumnya, maka nilai rata-rata prediksi probabilitas yang didapatkan adalah seperti yang ditunjukkan pada Tabel 8.

**Tabel 8** Rata-Rata Nilai Prediksi Probabilitas dengan 11 Pengulangan

| No | Pengulangan Ke- | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | ... | 11 |
| 1 | 0,4561 | 0,5709 | 0,4898 | 0,4946 | 0,5152 | 0,5252 | ... | 0,4894 |
| 2 | 0,8664 | 0,9198 | 0,7442 | 0,8263 | 0,7856 | 0,7402 | ... | 0,8140 |
| 3 | 0,5023 | 0,5286 | 0,5151 | 0,5425 | 0,4714 | 0,4186 | ... | 0,5393 |
| 4 | 0,8769 | 0,8357 | 0,8211 | 0,8356 | 0,8512 | 0,7966 | ... | 0,7458 |
| 5 | 0,7174 | 0,6401 | 0,5122 | 0,6661 | 0,6041 | 0,6007 | ... | 0,6080 |
| 6 | 0,2459 | 0,2491 | 0,3355 | 0,2166 | 0,2459 | 0,3416 | ... | 0,2376 |
| 7 | 0,2378 | 0,1837 | 0,3655 | 0,2562 | 0,3052 | 0,3956 | ... | 0,3724 |
| 8 | 0,3506 | 0,2462 | 0,3870 | 0,3741 | 0,4147 | 0,3041 | ... | 0,3621 |
| 9 | 0,5039 | 0,5871 | 0,5068 | 0,5317 | 0,4835 | 0,5237 | ... | 0,4917 |
| 10 | 0,2427 | 0,2388 | 0,3231 | 0,2563 | 0,3232 | 0,3538 | ... | 0,3398 |

Langkah 7: Mendapatkan prediksi kategori berdasarkan nilai rata-rata yang telah didapatkan pada Tabel 8 dengan *threshold* 0,5. Kemudian hasil akhir prediksi kategori ditentukan dengan cara majority voting. Hasil prediksi kategori yang didapatkan ditunjukkan dalam Tabel 9.

Prediksi pada setiap pengulangan didapatkan dengan kriteria jika prediksi probabilitas lebih dari 0,5, maka dimasukkan ke dalam kategori 1, dan sebaliknya untuk kategori 0. Setelah kategori prediksi didapatkan untuk seluruh pengulangan, selanjutnya dilakukan *voting* untuk kategori 1 dan 0, kemudian kategori prediksi

final yang digunakan sebagai *output* Lorens adalah merupakan hasil *voting* yang memiliki jumlah lebih besar.

**Tabel 4.9** Rata-Rata Nilai Prediksi Probabilitas dengan 11 Pengulangan

| No | Pengulangan Ke- | | | | | | | | | | | *Vote* | | Prediksi |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 0 | 1 | |
| 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 4 | 7 | 1 |
| 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 11 | 1 |
| 3 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 5 | 6 | 1 |
| 4 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 11 | 1 |
| 5 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 10 | 1 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 11 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 11 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 11 | 0 | 0 |
| 9 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 3 | 8 | 1 |
| 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 11 | 0 | 0 |

**Lampiran 2** Ilustrasi Analisis Data dengan Menggunakan Metode ELR

Ilustrasi analisis data menggunakan ELR dijelaskan pada uraian berikut ini. Langkah yang ditunjukkan disesuaikan dengan algoritma yang telah dijelaskan pada Tabel 2.3. Secara rinci, perhitungan pada analisis dengan menggunakan metode ELR dijelaskan dalam uraian berikut ini.

*Input* 1: Data sampel yang digunakan pada ilustrasi ini sama dengan yang dipakai pada ilustrasi dengan metode Lorens, yaitu yang ditunjukkan dalam Tabel 1 dalam Lampiran 1.

*Input* 2: Parameter regularisasi $\lambda$ yang digunakan dalam ilustrasi ini ditentukan sebesar 0,2.

*Input* 3: Inisialisasi vektor probabilitas berdasarkan perhitungan $t$-*test ranking*. Langkah perhitungan yang dilakukan berdasarkan Persamaan (2.18) pada variabel $x_1$ adalah sebagai berikut.

$$\hat{t}_1 = \frac{\hat{\mu}_{1+} - \hat{\mu}_{1-}}{\sqrt{\frac{\hat{\sigma}_{1+}^2}{n_+} + \frac{\hat{\sigma}_{1-}^2}{n_-}}} = \frac{0{,}420 - 0{,}652}{\sqrt{\frac{0{,}108}{5} + \frac{0{,}063}{5}}} = -1{,}255$$

Nilai yang digunakan sebagai probabilitas inisial adalah 1-$p$-*value*. Sehingga perlu dihitung nilai $p$-*value* untuk $t_1$ dengan derajat bebas atau $df$ berikut.

$$df = \frac{\sqrt{\frac{\hat{\sigma}_{1+}^2}{n_+} + \frac{\hat{\sigma}_{1-}^2}{n_-}}}{\frac{\left(\frac{\hat{\sigma}_{1+}^2}{n_+}\right)^2}{n_+ - 1} + \frac{\left(\frac{\hat{\sigma}_{1-}^2}{n_-}\right)^2}{n_- - 1}} = \frac{\sqrt{\frac{0{,}108}{5} + \frac{0{,}063}{5}}}{\frac{\left(\frac{0{,}108}{5}\right)^2}{5 - 1} + \frac{\left(\frac{0{,}063}{5}\right)^2}{5 - 1}} = \frac{0{,}00117}{0{,}000156} = 7{,}482$$

Dengan demikian, berdasarkan tabel distribusi $t$ dengan pengujian dua arah, maka $p$-*value* yang didapatkan adalah sebesar 0,247. Sehingga nilai probabilitasnya adalah sebesar 0,753. Dengan proses perhitungan yang sama, nilai probabilitas untuk seluruh variabel ditunjukkan dalam Tabel 10.

**Tabel 4.10** Nilai Inisial Probabilitas dari Perhitungan *t-Test Ranking*

| Variabel | Probabilitas | Variabel | Probabilitas |
|---|---|---|---|
| $x_1$ | 0,7520 | $x_6$ | 0,6029 |
| $x_2$ | 0,2779 | $x_7$ | 0,0077 |
| $x_3$ | 0,5893 | $x_8$ | 0,4295 |
| $x_4$ | 0,3652 | $x_9$ | 0,9311 |
| $x_5$ | 0,6780 | $x_{10}$ | 0,7154 |

*Input* 4: Inisialisasi $\overline{BCR}$ yang digunakan dalam ilustrasi ini adalah 0,5, karena jumlah data dalam kedua kategori seimbang.

Analisis untuk *ensemble* pertama dilakukan secara rinci seperti pada langkah berikut ini.

Langkah 1: Membuat partisi data *training* dan *testing*. Akan tetapi dalam ilustrasi ini tidak dilakukan partisi tersebut, sehingga data *testing* yang nantinya akan dicobakan merupakan data *training* yang digunakan untuk membentuk model.

Langkah 2: Mendapatkan $n$ dari $p$ variabel secara random berdasarkan inisialisasi probabilitas yang didapatkan dari *Input* 3 dengan tanpa pengembalian. Variabel-variabel yang terpilih pada langkah ini ditunjukkan dalam Tabel 11.

**Tabel 11** Variabel Terpilih secara Random Berdasarkan Inisialisasi Nilai Probabilitas

| No | Variabel Terpilih | Probabilitas |
|---|---|---|
| 1 | 1 | 0,7520 |
| 2 | 2 | 0,2779 |
| 3 | 3 | 0,5893 |
| 4 | 5 | 0,6780 |
| 5 | 6 | 0,6026 |
| 6 | 9 | 0,9311 |
| 7 | 10 | 0,7154 |
| 8 | 11 | 0,7412 |
| 9 | 13 | 0,5454 |
| 10 | 15 | 0,3023 |

Langkah 3: Membuat model regresi logistik biner terregularisasi menggunakan variabel terpilih yang didapatkan pada langkah sebelumnya. Hasil estimasi parameter yang didapatkan pada langkah ini ditunjukkan dalam Tabel 12.

**Tabel 12** Estimasi Parameter Regresi Logistik Terregularisasi

| Parameter | Estimasi | Parameter | Estimasi |
|---|---|---|---|
| $\beta_0$ | 1,7534 | $\beta_9$ | -2,1991 |
| $\beta_1$ | -0,6990 | $\beta_{10}$ | -1,3906 |
| $\beta_2$ | 0,1637 | $\beta_{11}$ | 0,6144 |
| $\beta_3$ | 0,4619 | $\beta_{13}$ | -0,5841 |
| $\beta_5$ | -0,6692 | $\beta_{15}$ | -0,3231 |
| $\beta_6$ | -0,3353 | | |

Langkah 4: Mendapatkan prediksi pada data *testing*. Hasil prediksi ditunjukkan pada Tabel 13. *Threshold* untuk menentukan kategori prediksi adalah 0,5.

**Tabel 13** Nilai Hasil Prediksi dari Model yang Didapatkan

| No | Prediksi Probabilitas | Prediksi Kategori |
|---|---|---|
| 1 | 0,6275 | 1 |
| 2 | 0,8626 | 1 |
| 3 | 0,4050 | 0 |
| 4 | 0,8919 | 1 |
| 5 | 0,7108 | 1 |
| 6 | 0,2457 | 0 |
| 7 | 0,2706 | 0 |
| 8 | 0,1374 | 0 |
| 9 | 0,4954 | 0 |
| 10 | 0,3533 | 0 |

Langkah 5: Menghitung nilai BCR. Tabel *confusion matrix* yang didapatkan berdasarkan prediksi pada Tabel 4.13 ditunjukkan dalam Tabel 4.14.

**Tabel 14** *Confusion Matrix* Data Ilustrasi

| | | Prediksi | |
|---|---|---|---|
| | | $P_P$ | $N_P$ |
| Aktual | $P_A$ | 4 | 1 |
| | $N_A$ | 0 | 5 |

Berdasarkan Persamaan (2.23), maka perhitungan nilai BCR adalah sebagai berikut.

$$BCR = \frac{1}{2}\left(\frac{TP}{P_A} + \frac{TN}{N_A}\right) = \frac{1}{2}\left(\frac{4}{5} + \frac{5}{5}\right) = 0{,}9$$

Langkah 6: Menghitung nilai *quality* berdasarkan Persamaan (2.22), seperti pada perhitungan di bawah ini.

$$quality = \log(1 + BCR_1 - \overline{BCR}_0) = \log(1 + 0{,}9 - 0{,}5) = 0{,}33647$$

Langkah 7: Memperbarui nilai vektor probabilitas berdasarkan Persamaan (2.21). Perhitungan untuk mendapatkan nilai probabilitas yang baru pada variabel pertama adalah sebagai berikut.

$$prob_{1,0} + quality.\beta_1^{2.sign(quality)} = \left(0.75204 + 0{,}33647(-0{,}69903)^{2(1)}\right)$$
$$= 0{,}91645$$

Dengan langkah yang sama, maka hasil perhitungan untuk seluruh variabel adalah sebagai berikut.

**Tabel 15** Nilai Hasil Prediksi dari Model yang Didapatkan

| Variabel | Hasil Perhitungan | Vektor Probabilitas yang Baru |
|---|---|---|
| $x_1$ | 0,9165 | 0,3582 |
| $x_2$ | 0,2870 | 0,1122 |
| $x_3$ | 0,6611 | 0,2584 |
| $x_5$ | 0,8287 | 0,3239 |
| $x_6$ | 0,6404 | 0,2503 |
| $x_9$ | 2,5583 | 1,0000 |
| $x_{10}$ | 1,3661 | 0,5340 |
| $x_{11}$ | 0,8682 | 0,3394 |
| $x_{13}$ | 0,6602 | 0,2581 |
| $x_{15}$ | 0,3375 | 0,1319 |

Vektor probabilitas yang baru dihasilkan dari normalisasi hasil perhitungan yang dicontohkan seperti perhitungan di atas. Angka untuk normalisasi ($z$) pada Persamaan (2.21) ditentukan dari nilai terbesar dari hasil perhitungan yang telah didapatkan, sehingga variabel terpilih yang memiliki hasil perhitungan paling besar akan memiliki nilai probabilitas sama dengan 1. Variabel yang tidak terpilih, probabilitasnya tidak diperbarui, sehingga sama dengan inisial probabilitas atau vektor probabilitas pada iterasi sebelumnya.

Langkah 8: Memperbarui nilai $\overline{BCR}$. Nilai $\overline{BCR}_1$ adalah sebagai berikut.

$$\overline{BCR}_1 = \frac{\overline{BCR}_0 + BCR_1}{2} = \frac{0{,}5 + 0{,}9}{2} = \frac{1{,}4}{2} = 0{,}7$$

Langkah 9: Menghitung nilai $\varepsilon$ untuk mengetahui apakah rangkaian analisis di atas menghasilkan nilai $\overline{BCR}$ yang sudah konvergen atau belum.

$\varepsilon = \overline{BCR}_0 - \overline{BCR}_1 = 0{,}5 - 0{,}7 = -0{,}2$

Berdasarkan perhitungan di atas, didapatkan bahwa nilai selisih $\overline{BCR}$ adalah -0,2. Jika kiriteria konvergensi pada algoritma ini ditentukan sebesar $10^{-5}$, maka didapatkan bahwa iterasi pertama pada algoritma ini masih belum konvergen. Dengan demikian, proses akan diulangi lagi mulai dari Langkah 1 hingga Langkah 9 dengan menggunakan vektor probabilitas dan nilai $\overline{BCR}$ yang baru.

**Lampiran 3** Program R untuk Membangkitkan Data Simulasi

```
simA = function(ndata, nvar.cont, mean, sigma, nvar.dis, prob, fixbeta=NULL,
pos.ratio, type.cont="uni", n.lc=0, beta.lc=NULL, tol.n=0) {
    iter = 0
    pos = NULL
    repeat {
        iter = iter + 1

        if (type.cont == "uni") {
            x.cont = matrix(ncol = nvar.cont, nrow = ndata)
            for (i in 1:nvar.cont) {
                x.cont[, i] = rnorm(ndata, mean = mean, sd = sigma)
            }
        } else if (type.cont == "multi") {
            require(mixtools)
            x.cont = rmvnorm(ndata, mean, sigma)
        } else if (type.cont == "lc") {
            x.cont = matrix(ncol = nvar.cont - n.lc, nrow = ndata)
            for (i in 1:(nvar.cont - n.lc)) {
                x.cont[, i] = rnorm(ndata, mean = mean, sd = sigma)
            }
            x.lc = matrix(ncol = n.lc, nrow = ndata)
            for (i in 1:n.lc) {
                x.lc[, i] = x.cont[, i:(i + length(beta.lc) - 1)] %*% beta.lc
            }
            x.cont = cbind(x.lc, x.cont)
        } else if (type.cont == "mlc") {
            require(mixtools)
            x.cont = rmvnorm(ndata, mean, sigma)
            x.lc = matrix(ncol = n.lc, nrow = ndata)
            for (i in 1:n.lc) {
                x.lc[, i] = x.cont[, i:(i + length(beta.lc) - 1)] %*% beta.lc
            }
            x.cont = cbind(x.lc, x.cont)
        }

        x.dis = matrix(ncol = nvar.dis, nrow = ndata)
        for (i in 1:nvar.dis) {
            x.dis[, i] = rbinom(ndata, size = 100, prob = prob)
        }
        x = cbind(x.cont, x.dis)

        sd = abs(rnorm(1, 5, 1))
        if (is.null(fixbeta)) beta = round(rnorm(nvar.cont + nvar.dis + 1, mean = 0,
sd = sd), 1) else beta = fixbeta

        ppos = 1 / (1 + exp(beta[1] + x %*% beta[-1]))
        y = vector(length = nrow(x))
        for (i in 1:nrow(x)) {
            if (ppos[i] > (1 - pos.ratio)) y[i] = 1 else y[i] = 0
            }
        pos[iter] = sum(y)
        taby = table(y)
        sim.data = cbind(y, x)

        cat("iter ", iter, ": ", pos[iter], "\n")

        break.cr = round(pos.ratio * ndata)
        int.break.cr = c((break.cr - tol.n):(break.cr + tol.n))

        if (length(which(int.break.cr == pos[iter])) > 0) break
        }

    return(list(data = sim.data, n.cat = taby, prob = ppos, beta = beta))
}
```

**Lampiran 4** Program R untuk Analisis Klasifikasi dengan Lorens

```
lr.cerp <- function(y,x,nens,fixsize=NULL,fixthres=NULL,search=F) {
        # initialization
        set.seed(as.numeric(Sys.time()))
        options(warn=-1)
        if(sum(is.na(x))>0) stop("missing value is found")
        if(sum(is.na(y))>0) stop("missing value is found")
        y <- as.data.frame(y)
        x <- as.data.frame(x)
        num_pred <- ncol(x)
        num_obs <- nrow(x)
        pos_rate <- sum(y)/num_obs

        # parameter search or default option
        if(search==T) {
                optimal <- search.thre_size(y,x,"lr")
                optsize <- optimal$size; opthreshold <- optimal$threshold
        }
        else {
                if(is.null(fixsize)) fixsize <- round(6*num_pred/num_obs)
                if(is.null(fixthres)) fixthres <- (pos_rate+.5)/2
                optsize <- fixsize; opthreshold <- fixthres
        }

        # main body
        ptss <- floor(seq(1,optsize+.999,length.out=num_pred))
        fitted <- NULL; predicted <- NULL; cname <- NULL; coef.table<-
matrix(0,num_pred,nens);
        partition.table <- matrix(0,num_pred,nens); intc <- matrix(0,optsize,nens);
probability <- rep(0,num_obs)
        for (i in 1:nens) {
                cname <- c(cname, paste("ens",i,sep=""))
                rand_pred <- sample(ptss)
                partition.table[,i] <- rand_pred
                avg_fit <- rep(0,num_obs)
for(j in 1:optsize) {
                        smp_dt <- cbind(y,x[,rand_pred==j])
                        intlr <- glm(y~.,data=smp_dt,family=binomial())
                        coef.vector <- intlr$coefficient
                        coef.vector[is.na(coef.vector)] <- 0
                        intc[j,i] <- coef.vector[1]; coef.vector <- coef.vector[-1]
                        coef.table[rand_pred==j,i] <- coef.vector
                        avg_fit <- avg_fit + intlr$fitted.values
                }
                fitted <- cbind(fitted,avg_fit/optsize,deparse.level=0)
                probability <- probability+(avg_fit/optsize)/nens
        }
        learning.decision <- ens.voting(fitted,opthreshold)$final.vote
        colnames(fitted) <- cname
        colnames(intc) <- cname
        colnames(coef.table) <- cname; rownames(coef.table) <- colnames(x)
        colnames(partition.table) <- cname; rownames(partition.table) <- colnames(x)

        return(list(fitted=fitted,probability=probability,learning.decision=learning.d
ecision,
                partition.table=partition.table,coef.table=coef.table,intercept=intc,

        number.ensemble=nens,optimal.size=optsize,optimal.threshold=opthre
shold))
}
```

**Lampiran 4** Program R untuk Analisis Klasifikasi dengan Lorens (Lanjutan)

```
### lr.cerp.predict applies lr.cerp model to new data(test set) similar as predict.lm
function.
### lr.cerp.object is required and built from lr.cerp function.
### xtest is also required and should be same format as x in lr.cerp function.
### ytest is optional if you want to check the accuracy
lr.cerp.predict <- function(lr.cerp.object,xtest,ytest=NULL) {
# initialization
        options(warn=-1)
        if(sum(is.na(xtest))>0) stop("missing value is found")
        if(sum(is.na(ytest))>0) stop("missing value is found")
        xtest <- as.data.frame(xtest)
        num_obs <- nrow(xtest)
        nens <- lr.cerp.object$number.ensemble
        optsize <- lr.cerp.object$optimal.size
        opthreshold <- lr.cerp.object$optimal.threshold

        # main body
        cname <- NULL; test.decision <- NULL; fitted <- NULL; probability <-
rep(0,num_obs)
        xtest <- xtest[,rownames(lr.cerp.object$partition.table)]
        for (i in 1:nens) {
                avg_fit <- rep(0,num_obs)
                cname <- c(cname, paste("ens",i,sep=""))
                curmod <- lr.cerp.object$partition.table[,i]
                for(j in 1:optsize) {
                        intc <- lr.cerp.object$intercept[j,i]
                        wrkmat <- xtest[,curmod==j]
                        cvec <- lr.cerp.object$coef.table[curmod==j,i]
                        int_vl <- as.matrix(wrkmat)%*%cvec
                        int_vl <- int_vl + intc
                        int_vl[int_vl>=709] <- 709
                        avg_fit <- avg_fit + exp(int_vl)/(1+exp(int_vl))
                }
                fitted <- cbind(fitted,avg_fit/optsize,deparse.level=0)
                probability <- probability+(avg_fit/optsize)/nens
        }
        test.decision <- ens.voting(fitted,opthreshold,ytest)
        colnames(fitted) <- cname

        return(list(fitted=fitted,probability=t(probability),decision=test.decision$fi
nal.vote,
optimal.size=optsize,optimal.threshold=opthreshold,decision.table=test.decision$twobyt
wo))
}

### lr.cerp.cv performs v-fold cross-validation using lr.cerp and lr.cerp.predict
functions.
### Options and requirements are the same as lr.cerp function.
### One additional requirement is v_fold which is the number of fold to be performed
for cross-validation.
lr.cerp.cv <- function(y,x,nens,v_fold,fixsize=NULL,fixthres=NULL,search=F) {
        # initialization
        set.seed(as.numeric(Sys.time()))
        options(warn=-1)
        if(sum(is.na(x))>0) stop("missing value is found")
        if(sum(is.na(y))>0) stop("missing value is found")
        y <- as.data.frame(y)
        x <- as.data.frame(x)
        num_obs <- nrow(y)
        rand_obs <- sample(1:num_obs)
        obs_rem <- num_obs%%v_fold
        obs_div <- (num_obs-obs_rem)/v_fold

        # main body
        probability <- rep(0,num_obs); predicted <- rep(0,num_obs); tbttable <-
matrix(0,2,2)
        part_size.list<-NULL; threshold.list<-NULL
```

**Lampiran 4** Program R untuk Analisis Klasifikasi dengan Lorens (Lanjutan)

```
        for(i in 1:v_fold) {
                if(i<=obs_rem) {head1<-(i-1)*(obs_div+1)+1;tail1<-i*(obs_div+1);}
                else {head1<-(i-1)*obs_div+obs_rem+1;tail1<-i*obs_div+obs_rem;}
                test_seq<-rand_obs[head1:tail1]
                learn_seq<-rand_obs[-c(head1:tail1)]
                ylearn<-y[learn_seq,];xlearn<-x[learn_seq,];xtest<-
x[test_seq,];ytest<-y[test_seq,]
                mid_rs<-lr.cerp(ylearn,xlearn,nens,fixsize,fixthres,search)
                pred_rs<-lr.cerp.predict(mid_rs,xtest,ytest)
                predicted[test_seq]<-pred_rs$decision
                for(j in 1:nens) probability[test_seq]<-
probability[test_seq]+pred_rs$fitted[,j]/nens
                tbttable<-tbttable+pred_rs$decision.table
                part_size.list<-c(part_size.list,mid_rs$optimal.size)
                threshold.list<-c(threshold.list,mid_rs$optimal.threshold)
        }

        return(
list(probability=probability,predicted=predicted,partition.size.list=part_size.list,
                threshold.list=threshold.list,decision.table=tbttable))
}

### internal functions
ens.voting <- function (tot_res,threshold,y=NULL) {
        nens<-ncol(tot_res);nobs<-nrow(tot_res)
        if (!is.null(y)) {real_pos<-sum(y);real_neg<-nobs-real_pos}
        tot_res[tot_res>=threshold] <- 1; tot_res[tot_res<threshold] <- 0
        final.vote <- rep(0,nobs)
        for(i in 1:nobs) final.vote[i] <- mean(tot_res[i,])
        final.vote[final.vote>=0.5] <- 1; final.vote[final.vote<0.5] <- 0
        twobytwo <- NULL
        if (!is.null(y)) {
                real_pred_pos <- sum(final.vote==y&y==1)
                real_pred_neg <- sum(final.vote==y&y==0)
                real_pos_pred_neg <- real_pos - real_pred_pos
                real_neg_pred_pos <- real_neg - real_pred_neg
                twobytwo <-
rbind(c(real_pred_pos,real_pos_pred_neg),c(real_neg_pred_pos,real_pred_neg))
                rownames(twobytwo) <- c("real.pos","real.neg")
                colnames(twobytwo) <- c("pred.pos","pred.neg")
        }
        return(list(final.vote=final.vote,twobytwo=twobytwo))
}

search.thre_size <- function (y,x,method) {
        nprd <- ncol(x);nobs <- nrow(x);orate <- sum(y)/nobs
        szseq <- NULL; int_fits <- NULL
        initseed <- c(2,3,4,5,6,7,8,9,10,12)
        for (i in initseed) {
                ipts<-i*nprd/nobs
                ipts<-floor(ipts)
                if (ipts%%2==0) ipts<-ipts+1
                if (szseq[length(szseq)]!=ipts||is.null(szseq)) {
                        szseq <- c(szseq,ipts)
                        int_fits <- cbind(int_fits,cv.fit(y,x,ipts,method))
                }
        }

        nsrsz <- length(szseq)
        add_fits<-NULL;addsz<-NULL
        if(orate>=.5) iseq<-seq(.5,orate,.02)
        else {iseq<-seq(.5,orate,-.02); iseq<-rev(iseq)}
        nbis<-length(iseq)
        szfth<-rep(0,nbis);acfth<-rep(0,nbis)
```

**Lampiran 4** Program R untuk Analisis Klasifikasi dengan Lorens (Lanjutan)

```
        for(j in 1:nbis) {
                acseq<-rep(0,nsrsz)
                for(k in 1:nsrsz) {
                        tmpf<-rep(0,nobs)
                        tmpf[int_fits[,k]>=iseq[j]]<-1;tmpf[int_fits[,k]<iseq[j]]<-0
                        acseq[k]<-sum(tmpf==y)/nobs
                }
                nbst<-sum(acseq==max(acseq));scol<-seq(1:nsrsz)
                if(nbst==1) nthc<-scol[acseq==max(acseq)]
                else {
                        tmpcol<-scol[acseq==max(acseq)]
                        nthc<-tmpcol[round(nbst/2)]
                }
                if(nthc==1) {
                        upts<-szseq[nthc+1];lpts<-szseq[nthc]
                        utfac<-acseq[nthc+1];ltfac<-acseq[nthc]
                        while(lpts!=upts) {
                                mpts<-(lpts+upts)/2
                                mpts<-floor(mpts)
                                if(mpts%%2==0) mpts<-mpts+1
                                if(mpts==upts) break
                                if(length(addsz)==0) {
                                        mtf<-cv.fit(y,x,mpts,method)
                                        addsz<-c(addsz,mpts);add_fits<-
cbind(add_fits,mtf)
                                }
                                else if(sum(addsz==mpts)==0) {
                                        mtf<-cv.fit(y,x,mpts,method)
                                        addsz<-c(addsz,mpts);add_fits<-
cbind(add_fits,mtf)
                                }
                                else mtf<-add_fits[,addsz==mpts]
                                tmtf<-rep(0,nobs)
                                tmtf[mtf>=iseq[j]]<-1;tmtf[mtf<iseq[j]]<-0
                                mtfac<-sum(tmpf==y)/nobs
                                if(ltfac>utfac) {
                                        if(mtfac>=utfac) {upts<-mpts;utfac<-mtfac}
                                        else {upts<-lpts;utfac<-ltfac}
                                }
                                else if(ltfac<utfac) {
                                        if(mtfac>=ltfac) {lpts<-mpts;ltfac<-mtfac}
                                        else {lpts<-upts;ltfac<-utfac}
                                }
                                else {
                                        if(mtfac>=ltfac) {
                                                lpts<-mpts;ltfac<-mtfac
                                                upts<-mpts;utfac<-mtfac
                                        }
                                        else {upts<-lpts;utfac<-ltfac}
                                }
                        }
                        if(ltfac>=utfac) {szfth[j]<-lpts;acfth[j]<-ltfac}
                        else {szfth[j]<-upts;acfth[j]<-utfac}
                }

                else if(nthc==nsrsz) {
                        lpts<-szseq[nthc-1];upts<-szseq[nthc]
                        ltfac<-acseq[nthc-1];utfac<-acseq[nthc]
                        while(lpts!=upts) {
                                mpts<-(lpts+upts)/2
                                mpts<-floor(mpts)
                                if(mpts%%2==0) mpts<-mpts+1
                                if(mpts==upts) break
                                if(length(addsz)==0) {
                                        mtf<-cv.fit(y,x,mpts,method)
                                        addsz<-c(addsz,mpts);add_fits<-
cbind(add_fits,mtf)
                                }
```

**Lampiran 4** Program R untuk Analisis Klasifikasi dengan Lorens (Lanjutan)

```
                                else if(sum(addsz==mpts)==0) {
                                        mtf<-cv.fit(y,x,mpts,method)
                                        addsz<-c(addsz,mpts);add_fits<-
cbind(add_fits,mtf)
                                }
                                else mtf<-add_fits[,addsz==mpts]
                                tmtf<-rep(0,nobs)
                                tmtf[mtf>=iseq[j]]<-1;tmtf[mtf<iseq[j]]<-0
                                mtfac<-sum(tmpf==y)/nobs
                                if(ltfac>utfac) {
                                        if(mtfac>=utfac) {upts<-mpts;utfac<-mtfac}
                                        else {upts<-lpts;utfac<-ltfac}
                                }
                                else if(ltfac<utfac) {
                                        if(mtfac>=ltfac) {lpts<-mpts;ltfac<-mtfac}
                                        else {lpts<-upts;ltfac<-utfac}
                                }
                                else {
                                        if(mtfac>=ltfac) {
                                                lpts<-mpts;ltfac<-mtfac
                                                upts<-mpts;utfac<-mtfac
                                        }
                                        else {upts<-lpts;utfac<-ltfac}
                                }
                        }
                        if(ltfac>=utfac) {szfth[j]<-lpts;acfth[j]<-ltfac}
                        else {szfth[j]<-upts;acfth[j]<-utfac}
                }

                else {
                        lpts<-szseq[nthc-1];upts<-szseq[nthc]
                        ltfac<-acseq[nthc-1];utfac<-acseq[nthc]
                        while(lpts!=upts) {
                                mpts<-(lpts+upts)/2
                                mpts<-floor(mpts)
                                if(mpts%%2==0) mpts<-mpts+1
                                if(mpts==upts) break
                                if(length(addsz)==0) {
                                        mtf<-cv.fit(y,x,mpts,method)
                                        addsz<-c(addsz,mpts);add_fits<-
cbind(add_fits,mtf)
                                }
                                else if(sum(addsz==mpts)==0) {
                                        mtf<-cv.fit(y,x,mpts,method)
                                        addsz<-c(addsz,mpts);add_fits<-
cbind(add_fits,mtf)
                                }
                                else mtf<-add_fits[,addsz==mpts]
                                tmtf<-rep(0,nobs)
                                tmtf[mtf>=iseq[j]]<-1;tmtf[mtf<iseq[j]]<-0
                                mtfac<-sum(tmpf==y)/nobs
                                else if(ltfac<utfac) {
                                        if(mtfac>=ltfac) {lpts<-mpts;ltfac<-mtfac}
                                        else {lpts<-upts;ltfac<-utfac}
                                }
                                else {
                                        if(mtfac>=ltfac) {
                                                lpts<-mpts;ltfac<-mtfac
                                                upts<-mpts;utfac<-mtfac
                                        }
                                        else {upts<-lpts;utfac<-ltfac}
                                }
                        }
                        if(ltfac>=utfac) {lsps<-lpts;lsbs<-ltfac}
                        else {lsps<-upts;lsbs<-utfac}
                        upts<-szseq[nthc+1];lpts<-szseq[nthc]
                        utfac<-acseq[nthc+1];ltfac<-acseq[nthc]
```

**Lampiran 4** Program R untuk Analisis Klasifikasi dengan Lorens (Lanjutan)

```
                        while(lpts!=upts) {
                                mpts<-(lpts+upts)/2
                                mpts<-floor(mpts)
                                if(mpts%%2==0) mpts<-mpts+1
                                if(mpts==upts) break
                                if(length(addsz)==0) {
                                        mtf<-cv.fit(y,x,mpts,method)
                                        addsz<-c(addsz,mpts);add_fits<-
cbind(add_fits,mtf)
                                }
                                else if(sum(addsz==mpts)==0) {
                                        mtf<-cv.fit(y,x,mpts,method)
                                        addsz<-c(addsz,mpts);add_fits<-
cbind(add_fits,mtf)
                                }
                                else mtf<-add_fits[,addsz==mpts]
                                tmtf<-rep(0,nobs)
                                tmtf[mtf>=iseq[j]]<-1;tmtf[mtf<iseq[j]]<-0
                                mtfac<-sum(tmpf==y)/nobs
                                if(ltfac>utfac) {
                                        if(mtfac>=utfac) {upts<-mpts;utfac<-mtfac}
                                        else {upts<-lpts;utfac<-ltfac}
                                }
                                else if(ltfac<utfac) {
                                        if(mtfac>=ltfac) {lpts<-mpts;ltfac<-mtfac}
                                        else {lpts<-upts;ltfac<-utfac}
                                }

                                else {
                                        if(mtfac>=ltfac) {
                                                lpts<-mpts;ltfac<-mtfac
                                                upts<-mpts;utfac<-mtfac
                                        }
                                        else {upts<-lpts;utfac<-ltfac}
                                }
                        }
                        if(ltfac>=utfac) {usps<-lpts;usbs<-ltfac}
                        else {usps<-upts;usbs<-utfac}
                        if(lsbs>=usbs) {szfth[j]<-lsps;acfth[j]<-lsbs}
                        else {szfth[j]<-usps;acfth[j]<-usbs}
                }
        }
        fnbst<-sum(max(acfth)==acfth);fscol<-seq(1:nbis)
        if(fnbst==1) {
                finsz<-szfth[max(acfth)==acfth]
                finth<-iseq[max(acfth)==acfth]
        }
        else {
                ftmpcol<-fscol[max(acfth)==acfth]
                tgcol<-ftmpcol[round(fnbst/2)]
                finsz<-szfth[tgcol]
                finth<-iseq[tgcol]
        }
        return(list(size=finsz,threshold=finth))
}

cv.fit <- function (y,x,npt,method) {
        num_pred<-ncol(x)
        num_obs<-nrow(x)
        lfit<-rep(0,num_obs)
        nv=3
        if(method=="lr") lfit<-lr.cerp.cv(y,x,1,nv)$probability
        else if(method=="lrt") lfit<-lrt.cerp.cv(y,x,1,nv)$probability
        else if(method=="ct") lfit<-ct.cerp.cv(y,x,1,nv)$probability
        return(lfit)
}
```

**Lampiran 5** Program R untuk Analisis Klasifikasi dengan ELR

```
base.model = function(ytrain, xtrain, ytest = NULL, xtest = NULL, alpha = 0,
fix.lambda = NULL, nlambda = 100, type.measure = "auc", nfolds.reg.lr = 3) {
    require(glmnet)
    if (is.null(fix.lambda)) {
        repeat {
            model = try(cv.glmnet(y = as.factor(ytrain), x = as.matrix(xtrain), family
= "binomial", alpha = alpha, nlambda = nlambda,
                    type.measure = type.measure, nfolds = nfolds.reg.lr, standardize =
FALSE),silent = TRUE)
            if (isTRUE(class(model) != "try-error")) break
            else print("Please wait...")
        }
        lambda=model$lambda.min
    }
    else {
        repeat {
            model = try(glmnet(y = as.factor(ytrain), x = as.matrix(xtrain), family =
"binomial", alpha = alpha, lambda = fix.lambda,
                    standardize = FALSE), silent = TRUE)
            if (isTRUE(class(model) != "try-error")) break
            else print("Please wait...")
        }
        lambda=model$lambda
    }

    coef = coef(model, s = "lambda.min")
    train.pred = as.factor(predict(model, as.matrix(xtrain), s = "lambda.min", type =
"class"))
    levels(train.pred) =levels(as.factor(ytrain))
    train.prob = predict(model, as.matrix(xtrain), s = "lambda.min", type =
"response")
    test.pred = as.factor(predict(model, as.matrix(xtest), s = "lambda.min", type =
"class"))
    levels(test.pred)=levels(as.factor(ytest))
    test.prob = predict(model, as.matrix(xtest), s = "lambda.min", type = "response")

    require(pROC)
    require(caret)
    conf.train = confusionMatrix(data = as.factor(train.pred), reference =
as.factor(ytrain))
    acc.train = conf.train$overall[1]
    bcr.train = conf.train$byClass[11]
    auc.train = pROC::roc(as.factor(ytrain), as.vector(as.numeric(train.prob)))$auc

    conf.test = confusionMatrix(data = as.factor(test.pred), reference =
as.factor(ytest))
    acc.test = conf.test$overall[1]
    bcr.test = conf.test$byClass[11]
    auc.test = pROC::roc(as.factor(ytest), as.vector(as.numeric(test.prob)))$auc

    return(list(model = model, coefficients = coef, training.data = cbind(y = ytrain,
xtrain), testing.data = cbind(y = ytest, xtest),
    test.prediction = test.pred, test.probability = test.prob, bcr = c(training.bcr =
bcr.train, testing.bcr = bcr.test),
    accuracy = c(training.accuracy = acc.train, testing.accuracy = acc.test), auc =
c(training.AUC = auc.train, testing.AUC = auc.test),
    lambda = lambda))
}

elr = function(y, x, prob.feature, tr.ratio = 8 / 10, alpha = 0, fix.lambda = NULL,
nlambda = 100, type.measure = "auc", nfolds.reg.lr = 3, tol = 1e-5) {
    n.feature = ncol(x)

    bcr.value = acc.value = auc.value = NULL
    quality = NULL
    avg.bcr = NULL
    cfeat = NULL
```

**Lampiran 5** Program R untuk Analisis Klasifikasi dengan ELR (Lanjutan)

```
     avg.bcr[1] = length(which(y == 1)) / length(y)
    prob = matrix(nrow = length(prob.feature))
    prob[, 1] = prob.feature

    iter = 1
    bcr.value = auc.value = acc.value = feature = NULL
    repeat {
        iter = iter + 1
        cat("iter", iter, "\n")
        repeat {
            id0 = which(y == 0)
            id1 = which(y == 1)
            id0.train = sample(id0, round(tr.ratio * length(id0)), replace = F)
            id1.train = sample(id1, round(tr.ratio * length(id1)), replace = F)
            match.id0 = match(id0.train, id0)
            match.id1 = match(id1.train, id1)
            dtrain.pos = x[id1[match.id1],]
            dtrain.neg = x[id0[match.id0],]
            dtest.pos = x[id1[-match.id1],]
            dtest.neg = x[id0[-match.id0],]
            ttrain.pos = y[id1[match.id1]]
            ttrain.neg = y[id0[match.id0]]
            ttest.pos = y[id1[-match.id1]]
            ttest.neg = y[id0[-match.id0]]

            chs.feature = sort(sample(x = c(1:n.feature), size = nrow(dtrain.pos) +
nrow(dtrain.neg), replace = FALSE, prob = prob[, iter - 1]))

            dtrain = rbind(dtrain.neg[, chs.feature], dtrain.pos[, chs.feature])
            dtest = rbind(dtest.neg[, chs.feature], dtest.pos[, chs.feature])
            ttrain = c(ttrain.neg, ttrain.pos)
            ttest = c(ttest.neg, ttest.pos)

            model = base.model(ytrain = as.vector(ttrain), xtrain = as.matrix(dtrain),
                    ytest = as.vector(ttest), xtest = as.matrix(dtest), alpha = alpha,
fix.lambda=NULL, nlambda = nlambda,
                    type.measure = type.measure, nfolds.reg.lr = nfolds.reg.lr)

            coef.model = coef(model, s = "lambda.min")
            coef.m = coef.model
            cfeat = chs.feature
            lambda = model$lambda
            test.pred = model$test.prediction
            test.prob = model$test.probability

            quality[iter] = log10(1 + abs(model$bcr[2] - avg.bcr[iter - 1]))
            upd.prob = prob[chs.feature] + quality[iter] * as.vector(coef.model)[-1] ^
(2 * sign(quality[iter]))
            upd.prob[which(upd.prob == -Inf | upd.prob == Inf)] = 0
            prob = cbind(prob, rep(NA, length(prob.feature)))
            prob[chs.feature, iter] = abs(upd.prob) /
abs(upd.prob[which.max(abs(upd.prob))[1]])
            prob[-chs.feature, iter] = prob[-chs.feature, (iter - 1)]
            avg.bcr[iter] = (((iter - 1) * avg.bcr[iter - 1]) + model$bcr[2]) / iter

            if (length(which(prob[, iter] < 0)) == 0) break
            }
        bcr.value = rbind(bcr.value, model$bcr)
        acc.value = rbind(acc.value, model$accuracy)
        auc.value = rbind(auc.value, model$auc)
        feature=cbind(feature,cfeat)

        eps = abs(avg.bcr[iter - 1] - avg.bcr[iter])
        if (eps <= tol & iter > 10) break
        }
```

**Lampiran 5** Program R untuk Analisis Klasifikasi dengan ELR (Lanjutan)

```
    return(list(quality=quality, upd.prob=prob, model = model, bcr = bcr.value,
accuracy = acc.value, auc = auc.value, average.bcr = avg.bcr, chosen.feature =
feature,
    coefficients = coef.m, lambda = lambda, test.prediction = test.pred,
test.probability = test.prob, test.target=ttest, last.prob=prob[,iter-1]))
}
```

**Lampiran 6** Data *Drug Discovery* mengenai Radioproteksi Sel

| Index | Name | Target | pKa(max20) | Br_Count | C_Count | ⋯ | Molecular_Volume |
|---|---|---|---|---|---|---|---|
| 0 | AS-1 | 1 | 20 | 0 | 19 | ⋯ | 351.91 |
| 1 | AS-10 | 1 | 11.4 | 0 | 11 | ⋯ | 224.66 |
| 2 | AS-11 | 0 | 7.9 | 0 | 10 | ⋯ | 170.81 |
| 3 | AS-12 | 1 | 8.3 | 0 | 11 | ⋯ | 188.3 |
| 4 | AS-13 | 0 | 20 | 0 | 11 | ⋯ | 140.62 |
| 5 | AS-15 | 1 | 20 | 0 | 12 | ⋯ | 202.36 |
| 6 | AS-16 | 1 | 10.1 | 0 | 13 | ⋯ | 261.02 |
| 7 | AS-17 | 1 | 20 | 0 | 13 | ⋯ | 288.11 |
| 8 | AS-2 | 1 | 11.5 | 0 | 12 | ⋯ | 244.9 |
| 9 | AS-3 | 0 | 20 | 0 | 10 | ⋯ | 139.6 |
| 10 | AS-4 | 0 | 20 | 0 | 11 | ⋯ | 154.69 |
| 11 | AS-5 | 1 | 20 | 0 | 13 | ⋯ | 222.94 |
| 12 | AS-6 | 1 | 20 | 0 | 14 | ⋯ | 234.95 |
| 13 | AS-7 | 1 | 6.4 | 0 | 12 | ⋯ | 228.43 |
| 14 | AS-8 | 1 | 20 | 0 | 16 | ⋯ | 311.78 |
| 15 | AS-9 | 0 | 20 | 0 | 20 | ⋯ | 369.41 |
| 16 | KH-1 | 1 | 9.8 | 0 | 9 | ⋯ | 108.73 |
| 17 | KH-10 | 0 | 10.1 | 0 | 10 | ⋯ | 130.68 |
| 18 | KH-12 | 0 | 8.2 | 0 | 12 | ⋯ | 207.17 |
| 19 | KH-13 | 1 | 7 | 0 | 14 | ⋯ | 278.85 |
| 20 | KH-16 | 0 | 7.2 | 0 | 18 | ⋯ | 337.85 |
| 21 | KH-18 | 1 | 20 | 0 | 15 | ⋯ | 297.03 |
| 22 | KH-19 | 0 | 6.3 | 1 | 12 | ⋯ | 222.94 |
| 23 | KH-2 | 1 | 9.4 | 0 | 9 | ⋯ | 116.27 |
| 24 | KH-20 | 1 | 20 | 1 | 13 | ⋯ | 239.41 |
| 25 | KH-21 | 0 | 20 | 1 | 14 | ⋯ | 251.76 |
| 26 | KH-22 | 1 | 20 | 0 | 14 | ⋯ | 251.76 |
| 27 | KH-23 | 0 | 20 | 0 | 14 | ⋯ | 261.36 |
| 28 | KH-24 | 1 | 10.5 | 0 | 20 | ⋯ | 377.64 |
| 29 | KH-25 | 1 | 4.8 | 0 | 12 | ⋯ | 227.06 |
| 30 | KH-3 | 1 | 9 | 0 | 9 | ⋯ | 129.31 |
| 31 | KH-4 | 1 | 9 | 1 | 9 | ⋯ | 134.45 |
| 32 | KH-5 | 0 | 9.1 | 0 | 9 | ⋯ | 143.03 |
| 33 | KH-6 | 0 | 7.8 | 0 | 11 | ⋯ | 186.24 |
| 34 | KT-1 | 0 | 11.9 | 0 | 14 | ⋯ | 293.95 |
| 35 | KT-2 | 1 | 20 | 0 | 18 | ⋯ | 333.73 |
| 36 | MH-1 | 1 | 6.8 | 0 | 16 | ⋯ | 240.44 |
| 37 | MH-10 | 0 | 20 | 0 | 14 | ⋯ | 230.15 |
| 38 | MH-11 | 0 | 20 | 0 | 15 | ⋯ | 252.44 |
| 39 | MH-12 | 0 | 20 | 0 | 14 | ⋯ | 232.55 |
| 40 | MH-13 | 1 | 7.7 | 0 | 18 | ⋯ | 332.02 |
| 41 | MH-14 | 0 | 20 | 0 | 19 | ⋯ | 350.88 |
| 42 | MH-15 | 1 | 20 | 0 | 10 | ⋯ | 137.19 |
| 43 | MH-16 | 1 | 11.8 | 0 | 10 | ⋯ | 135.48 |
| 44 | MH-2 | 0 | 6.8 | 0 | 18 | ⋯ | 264.79 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋱ | ⋮ |
| 80 | YN-7 | 1 | 20 | 0 | 14 | ⋯ | 315.21 |
| 81 | YN-8 | 0 | 20 | 0 | 12 | ⋯ | 193.1 |
| 82 | YN-9 | 1 | 20 | 0 | 15 | ⋯ | 306.98 |
| 83 | YT-1 | 1 | 9.8 | 0 | 20 | ⋯ | 363.23 |

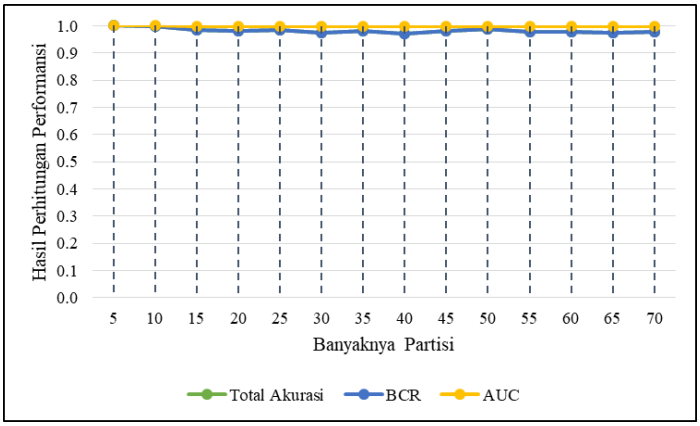
**Lampiran 7** Data Simulasi BMV-111

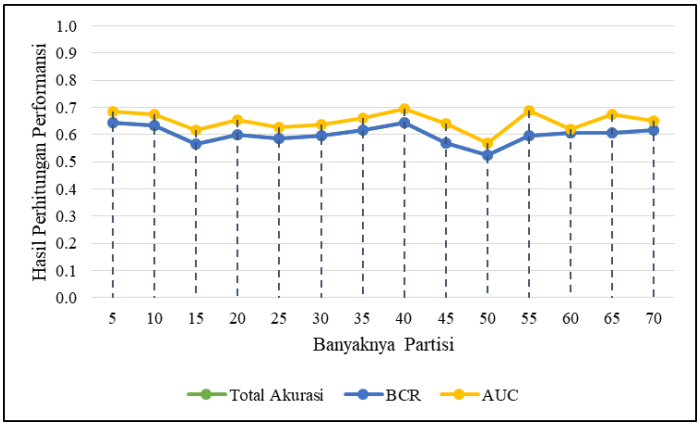
| No | y | x1 | x2 | x3 | x4 | x5 | ⋯ | x160 | x161 | x162 | x163 | x164 | ⋯ | x200 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 5.4583 | 4.2774 | 4.4965 | 5.8866 | 5.9106 | ⋯ | 3.4513 | 81 | 74 | 79 | 80 | ⋯ | 83 |
| 2 | 0 | 4.8458 | 5.2470 | 4.3444 | 4.1335 | 7.0375 | ⋯ | 5.1652 | 89 | 78 | 77 | 77 | ⋯ | 90 |
| 3 | 0 | 5.7346 | 5.0185 | 5.6780 | 5.8630 | 4.5434 | ⋯ | 3.3532 | 82 | 83 | 81 | 77 | ⋯ | 80 |
| 4 | 0 | 5.0162 | 5.0481 | 5.6099 | 5.3062 | 5.4161 | ⋯ | 6.6600 | 81 | 85 | 87 | 87 | ⋯ | 80 |
| 5 | 1 | 4.9758 | 4.7878 | 5.0580 | 5.7885 | 5.1812 | ⋯ | 5.5084 | 74 | 79 | 78 | 83 | ⋯ | 84 |
| 6 | 1 | 6.2109 | 4.9835 | 4.9409 | 4.1677 | 5.2379 | ⋯ | 5.1559 | 84 | 81 | 70 | 87 | ⋯ | 86 |
| 7 | 1 | 5.7909 | 3.9843 | 6.3767 | 3.6155 | 5.2482 | ⋯ | 4.4299 | 72 | 80 | 79 | 87 | ⋯ | 76 |
| 8 | 0 | 5.1858 | 5.9052 | 5.7216 | 6.7026 | 3.7472 | ⋯ | 6.2898 | 75 | 84 | 84 | 71 | ⋯ | 72 |
| 9 | 0 | 5.5909 | 6.4837 | 6.5446 | 5.0064 | 4.7404 | ⋯ | 4.4478 | 86 | 80 | 87 | 81 | ⋯ | 79 |
| 10 | 1 | 6.0898 | 5.4478 | 3.9759 | 4.9313 | 5.4719 | ⋯ | 4.9316 | 84 | 82 | 85 | 80 | ⋯ | 84 |
| 11 | 1 | 6.4979 | 5.7036 | 5.1199 | 5.5862 | 3.9200 | ⋯ | 4.7263 | 83 | 74 | 81 | 86 | ⋯ | 80 |
| 12 | 0 | 5.5351 | 6.4446 | 5.8598 | 5.1530 | 4.7784 | ⋯ | 4.7710 | 78 | 80 | 79 | 76 | ⋯ | 84 |
| 13 | 1 | 5.2441 | 5.6396 | 6.3606 | 5.9189 | 6.5309 | ⋯ | 5.5396 | 82 | 87 | 73 | 81 | ⋯ | 79 |
| 14 | 1 | 6.1523 | 6.1713 | 3.4003 | 4.2325 | 5.8364 | ⋯ | 5.7013 | 82 | 80 | 77 | 78 | ⋯ | 77 |
| 15 | 1 | 3.8672 | 4.7775 | 4.7013 | 4.7006 | 3.3584 | ⋯ | 4.1614 | 88 | 76 | 77 | 84 | ⋯ | 87 |
| 16 | 0 | 5.5370 | 4.2303 | 5.6073 | 3.3363 | 5.8112 | ⋯ | 4.1079 | 81 | 77 | 88 | 74 | ⋯ | 83 |
| 17 | 1 | 5.4685 | 4.6414 | 4.5231 | 4.7266 | 5.1496 | ⋯ | 5.3709 | 76 | 78 | 75 | 82 | ⋯ | 77 |
| 18 | 0 | 4.5405 | 5.7623 | 5.6924 | 4.3666 | 5.2418 | ⋯ | 3.6646 | 74 | 83 | 74 | 74 | ⋯ | 79 |
| 19 | 0 | 6.3410 | 4.7932 | 5.5890 | 4.3628 | 4.5082 | ⋯ | 4.1980 | 82 | 65 | 81 | 83 | ⋯ | 81 |
| 20 | 0 | 4.2980 | 4.5213 | 5.7050 | 4.3685 | 5.0689 | ⋯ | 6.0909 | 70 | 79 | 86 | 76 | ⋯ | 80 |
| 21 | 1 | 5.5919 | 4.9151 | 5.2023 | 5.0275 | 6.9238 | ⋯ | 4.6869 | 84 | 78 | 74 | 79 | ⋯ | 81 |
| 22 | 0 | 3.6105 | 5.7626 | 3.9303 | 5.6394 | 5.3774 | ⋯ | 5.7247 | 80 | 87 | 78 | 79 | ⋯ | 86 |
| 23 | 0 | 4.8628 | 5.3766 | 5.7464 | 4.8824 | 4.7700 | ⋯ | 5.2116 | 79 | 87 | 89 | 81 | ⋯ | 84 |
| 24 | 0 | 3.9749 | 6.5541 | 4.9650 | 4.2309 | 5.8552 | ⋯ | 5.7142 | 74 | 83 | 86 | 76 | ⋯ | 81 |
| 25 | 1 | 5.1574 | 3.9831 | 6.0338 | 3.9000 | 6.2369 | ⋯ | 4.3897 | 79 | 78 | 78 | 81 | ⋯ | 79 |
| 26 | 1 | 5.7864 | 6.7492 | 5.0384 | 5.1192 | 4.4338 | ⋯ | 4.9229 | 79 | 82 | 79 | 77 | ⋯ | 81 |
| 27 | 0 | 5.5132 | 4.8653 | 5.2935 | 5.1337 | 5.3101 | ⋯ | 5.4306 | 77 | 80 | 88 | 79 | ⋯ | 77 |
| 28 | 1 | 5.5236 | 5.2706 | 4.7690 | 5.8385 | 5.7734 | ⋯ | 6.1680 | 84 | 84 | 82 | 80 | ⋯ | 79 |
| 29 | 0 | 5.6873 | 3.9464 | 4.1994 | 4.3180 | 3.9778 | ⋯ | 6.5492 | 74 | 87 | 83 | 74 | ⋯ | 80 |
| 30 | 1 | 4.9690 | 5.1951 | 4.5918 | 4.3373 | 4.1221 | ⋯ | 5.4823 | 80 | 76 | 79 | 78 | ⋯ | 92 |
| 31 | 1 | 4.2771 | 3.8352 | 5.0254 | 4.8706 | 5.6973 | ⋯ | 6.0708 | 76 | 84 | 80 | 82 | ⋯ | 85 |
| 32 | 0 | 4.4191 | 6.3072 | 6.7121 | 5.9717 | 4.8795 | ⋯ | 5.3488 | 81 | 76 | 86 | 78 | ⋯ | 85 |
| 33 | 0 | 4.1666 | 4.7231 | 6.0373 | 4.2018 | 4.8267 | ⋯ | 4.8319 | 78 | 84 | 80 | 77 | ⋯ | 83 |
| 34 | 1 | 5.4959 | 4.4546 | 4.5560 | 4.6592 | 5.8821 | ⋯ | 5.0902 | 69 | 81 | 80 | 76 | ⋯ | 83 |
| 35 | 1 | 5.1053 | 4.5499 | 3.3405 | 4.6960 | 5.5210 | ⋯ | 4.2269 | 77 | 78 | 78 | 82 | ⋯ | 80 |
| 36 | 1 | 4.5827 | 4.6016 | 6.2599 | 3.9634 | 4.4728 | ⋯ | 3.8211 | 73 | 74 | 81 | 85 | ⋯ | 75 |
| 37 | 1 | 4.5443 | 5.2398 | 5.8520 | 5.1250 | 4.8639 | ⋯ | 4.7419 | 74 | 70 | 81 | 82 | ⋯ | 75 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋱ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋱ | ⋮ |
| 100 | 0 | 5.1935 | 4.2666 | 6.1004 | 4.5683 | 5.9969 | ⋯ | 5.1227 | 81 | 83 | 71 | 71 | ⋯ | 84 |

**Lampiran 8** Hasil Analisis Simulasi Lorens

a. Skenario BMV 111

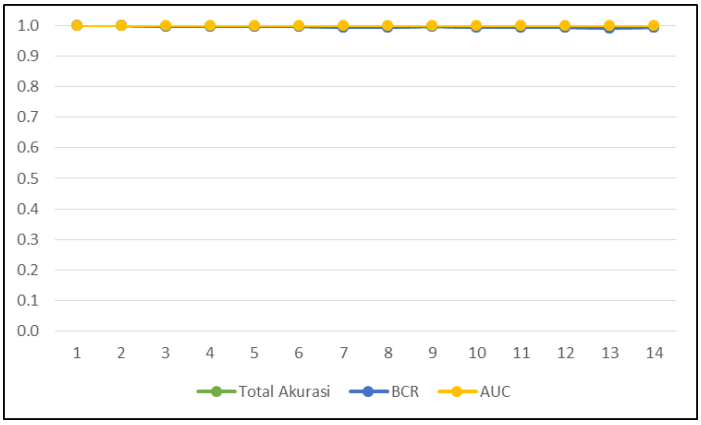
| Partisi | Data *Training* | | | Data *Testing* | | |
|---|---|---|---|---|---|---|
| | Total Akurasi | BCR | AUC | Total Akurasi | BCR | AUC |
| 5 | **1.0000** | **1.0000** | **1.0000** | **0.6450** | **0.6450** | 0.6860 |
| 10 | 0.9975 | 0.9975 | **1.0000** | 0.6350 | 0.6350 | 0.6730 |
| 15 | 0.9850 | 0.9850 | 0.9997 | 0.5650 | 0.5650 | 0.6160 |
| 20 | 0.9813 | 0.9813 | 0.9984 | 0.6000 | 0.6000 | 0.6550 |
| 25 | 0.9838 | 0.9838 | 0.9988 | 0.5850 | 0.5850 | 0.6250 |
| 30 | 0.9725 | 0.9725 | 0.9983 | 0.5950 | 0.5950 | 0.6360 |
| 35 | 0.9813 | 0.9813 | 0.9980 | 0.6150 | 0.6150 | 0.6620 |
| 40 | 0.9700 | 0.9700 | 0.9985 | **0.6450** | **0.6450** | **0.6950** |
| 45 | 0.9813 | 0.9813 | 0.9984 | 0.5700 | 0.5700 | 0.6400 |
| 50 | 0.9888 | 0.9888 | 0.9989 | 0.5250 | 0.5250 | 0.5690 |
| 55 | 0.9788 | 0.9788 | 0.9984 | 0.5950 | 0.5950 | 0.6880 |
| 60 | 0.9788 | 0.9788 | 0.9975 | 0.6050 | 0.6050 | 0.6190 |
| 65 | 0.9738 | 0.9738 | 0.9972 | 0.6050 | 0.6050 | 0.6750 |
| 70 | 0.9775 | 0.9775 | 0.9978 | 0.6150 | 0.6150 | 0.6510 |

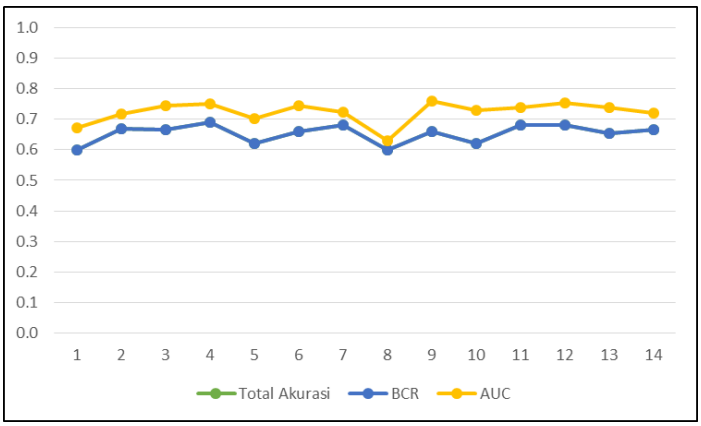


b. Skenario BMV-112

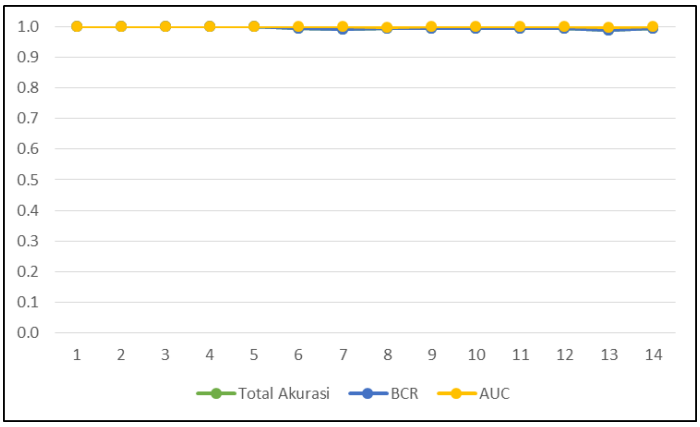
| Partisi | Data *Training* | | | Data *Testing* | | |
|---|---|---|---|---|---|---|
| | Total Akurasi | BCR | AUC | Total Akurasi | BCR | AUC |
| 5 | 1.0000 | 1.0000 | 1.0000 | 0.6000 | 0.6000 | 0.6730 |
| 10 | 1.0000 | 1.0000 | 1.0000 | 0.6700 | 0.6700 | 0.7170 |
| 15 | 0.9988 | 0.9988 | 1.0000 | 0.6650 | 0.6650 | 0.7450 |
| 20 | 0.9975 | 0.9975 | 1.0000 | 0.6900 | 0.6900 | 0.7520 |
| 25 | 0.9975 | 0.9975 | 1.0000 | 0.6200 | 0.6200 | 0.7010 |
| 30 | 0.9988 | 0.9988 | 1.0000 | 0.6600 | 0.6600 | 0.7440 |
| 35 | 0.9950 | 0.9950 | 0.9999 | 0.6800 | 0.6800 | 0.7240 |
| 40 | 0.9950 | 0.9950 | 1.0000 | 0.6000 | 0.6000 | 0.6290 |
| 45 | 0.9975 | 0.9975 | 0.9999 | 0.6600 | 0.6600 | 0.7600 |
| 50 | 0.9950 | 0.9950 | 1.0000 | 0.6200 | 0.6200 | 0.7290 |
| 55 | 0.9938 | 0.9938 | 0.9998 | 0.6800 | 0.6800 | 0.7370 |
| 60 | 0.9963 | 0.9963 | 0.9998 | 0.6800 | 0.6800 | 0.7550 |
| 65 | 0.9925 | 0.9925 | 0.9998 | 0.6550 | 0.6550 | 0.7380 |
| 70 | 0.9950 | 0.9950 | 0.9999 | 0.6650 | 0.6650 | 0.7190 |

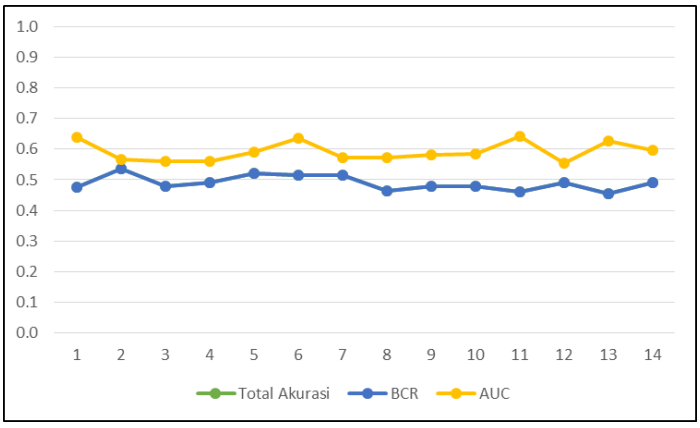


c. Skenario BMV-113

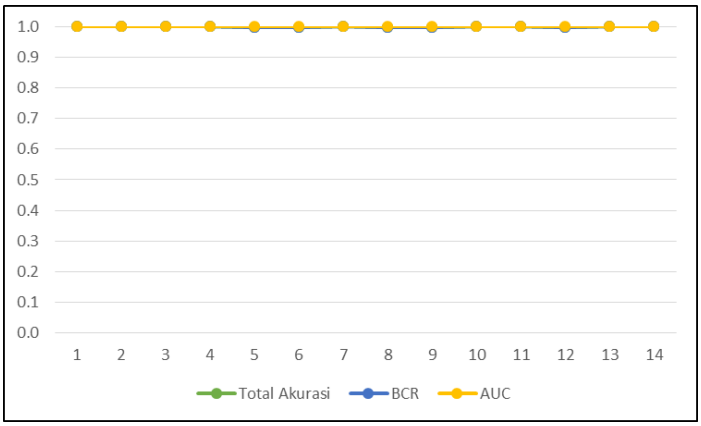
| Partisi | Data *Training* | | | Data *Testing* | | |
|---|---|---|---|---|---|---|
| | Total Akurasi | BCR | AUC | Total Akurasi | BCR | AUC |
| 5 | 1.0000 | 1.0000 | 1.0000 | 0.4750 | 0.4750 | 0.6390 |
| 10 | 1.0000 | 1.0000 | 1.0000 | 0.5350 | 0.5350 | 0.5660 |
| 15 | 1.0000 | 1.0000 | 1.0000 | 0.4800 | 0.4800 | 0.5600 |
| 20 | 1.0000 | 1.0000 | 1.0000 | 0.4900 | 0.4900 | 0.5600 |
| 25 | 1.0000 | 1.0000 | 1.0000 | 0.5200 | 0.5200 | 0.5920 |
| 30 | 0.9950 | 0.9950 | 1.0000 | 0.5150 | 0.5150 | 0.6360 |
| 35 | 0.9925 | 0.9925 | 0.9999 | 0.5150 | 0.5150 | 0.5720 |
| 40 | 0.9938 | 0.9938 | 0.9997 | 0.4650 | 0.4650 | 0.5710 |
| 45 | 0.9950 | 0.9950 | 0.9999 | 0.4800 | 0.4800 | 0.5800 |
| 50 | 0.9963 | 0.9963 | 0.9999 | 0.4800 | 0.4800 | 0.5850 |
| 55 | 0.9963 | 0.9963 | 0.9999 | 0.4600 | 0.4600 | 0.6430 |
| 60 | 0.9938 | 0.9938 | 0.9998 | 0.4900 | 0.4900 | 0.5540 |
| 65 | 0.9900 | 0.9900 | 0.9996 | 0.4550 | 0.4550 | 0.6260 |
| 70 | 0.9938 | 0.9938 | 1.0000 | 0.4900 | 0.4900 | 0.5970 |

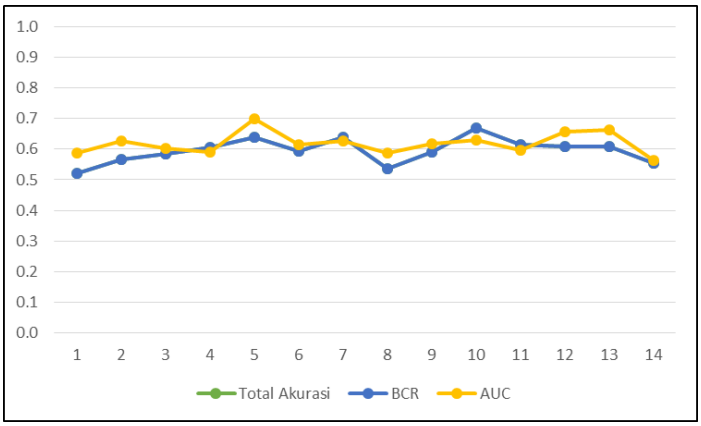


d. Skenario BMV-114

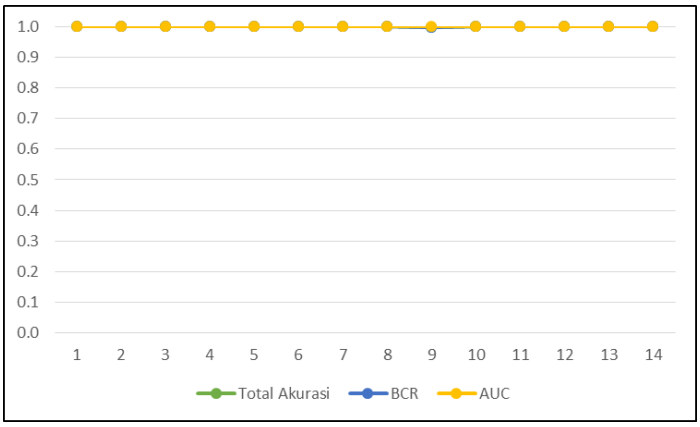
| Partisi | Data *Training* | | | Data *Testing* | | |
|---|---|---|---|---|---|---|
| | Total Akurasi | BCR | AUC | Total Akurasi | BCR | AUC |
| 5 | 1.0000 | 1.0000 | 1.0000 | 0.5200 | 0.5200 | 0.5880 |
| 10 | 1.0000 | 1.0000 | 1.0000 | 0.5650 | 0.5650 | 0.6260 |
| 15 | 1.0000 | 1.0000 | 1.0000 | 0.5850 | 0.5850 | 0.6040 |
| 20 | 1.0000 | 1.0000 | 1.0000 | 0.6050 | 0.6050 | 0.5900 |
| 25 | 0.9988 | 0.9988 | 1.0000 | 0.6400 | 0.6400 | 0.6980 |
| 30 | 0.9988 | 0.9988 | 1.0000 | 0.5950 | 0.5950 | 0.6160 |
| 35 | 1.0000 | 1.0000 | 1.0000 | 0.6400 | 0.6400 | 0.6260 |
| 40 | 0.9988 | 0.9988 | 1.0000 | 0.5350 | 0.5350 | 0.5870 |
| 45 | 0.9975 | 0.9975 | 1.0000 | 0.5900 | 0.5900 | 0.6180 |
| 50 | 1.0000 | 1.0000 | 1.0000 | 0.6700 | 0.6700 | 0.6300 |
| 55 | 1.0000 | 1.0000 | 1.0000 | 0.6150 | 0.6150 | 0.5960 |
| 60 | 0.9988 | 0.9988 | 1.0000 | 0.6100 | 0.6100 | 0.6560 |
| 65 | 1.0000 | 1.0000 | 1.0000 | 0.6100 | 0.6100 | 0.6630 |
| 70 | 1.0000 | 1.0000 | 1.0000 | 0.5550 | 0.5550 | 0.5620 |

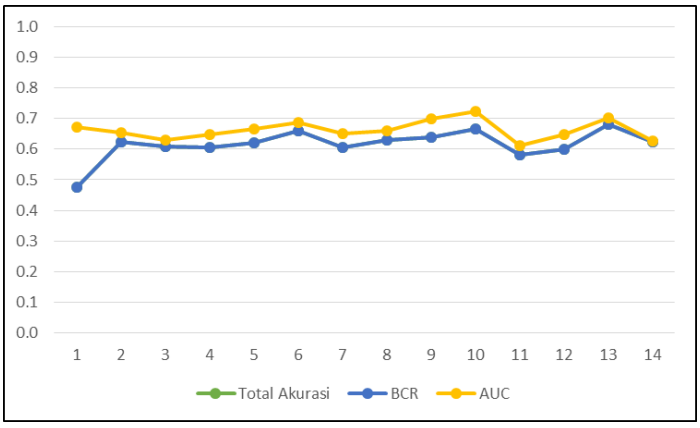


e. Skenario BMV-115

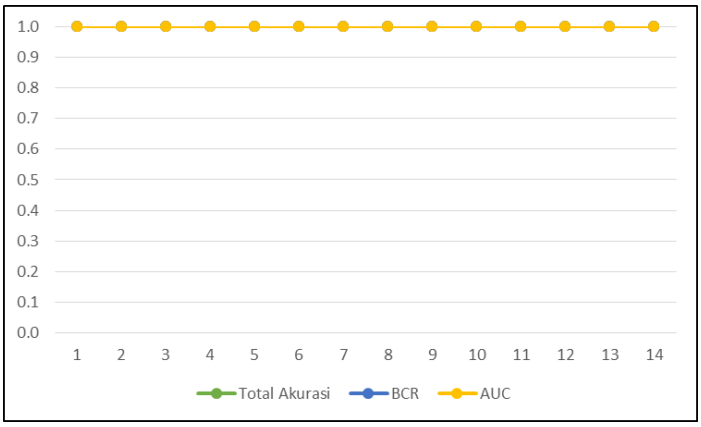
| Partisi | Data *Training* | | | Data *Testing* | | |
|---|---|---|---|---|---|---|
| | Total Akurasi | BCR | AUC | Total Akurasi | BCR | AUC |
| 5 | 1.0000 | 1.0000 | 1.0000 | 0.4750 | 0.4750 | 0.6730 |
| 10 | 1.0000 | 1.0000 | 1.0000 | 0.6250 | 0.6250 | 0.6550 |
| 15 | 1.0000 | 1.0000 | 1.0000 | 0.6100 | 0.6100 | 0.6300 |
| 20 | 1.0000 | 1.0000 | 1.0000 | 0.6050 | 0.6050 | 0.6490 |
| 25 | 1.0000 | 1.0000 | 1.0000 | 0.6200 | 0.6200 | 0.6650 |
| 30 | 1.0000 | 1.0000 | 1.0000 | 0.6600 | 0.6600 | 0.6860 |
| 35 | 1.0000 | 1.0000 | 1.0000 | 0.6050 | 0.6050 | 0.6500 |
| 40 | 1.0000 | 1.0000 | 1.0000 | 0.6300 | 0.6300 | 0.6610 |
| 45 | 0.9988 | 0.9988 | 1.0000 | 0.6400 | 0.6400 | 0.6980 |
| 50 | 1.0000 | 1.0000 | 1.0000 | 0.6650 | 0.6650 | 0.7240 |
| 55 | 1.0000 | 1.0000 | 1.0000 | 0.5800 | 0.5800 | 0.6130 |
| 60 | 1.0000 | 1.0000 | 1.0000 | 0.6000 | 0.6000 | 0.6470 |
| 65 | 1.0000 | 1.0000 | 1.0000 | 0.6800 | 0.6800 | 0.7020 |
| 70 | 1.0000 | 1.0000 | 1.0000 | 0.6250 | 0.6250 | 0.6270 |

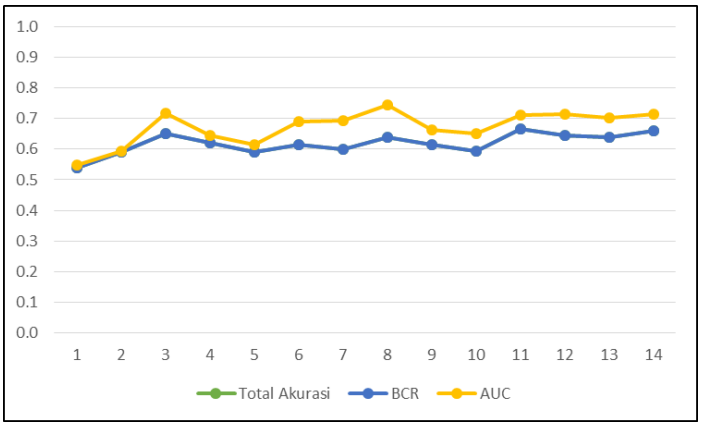


f. Skenario BMV-116

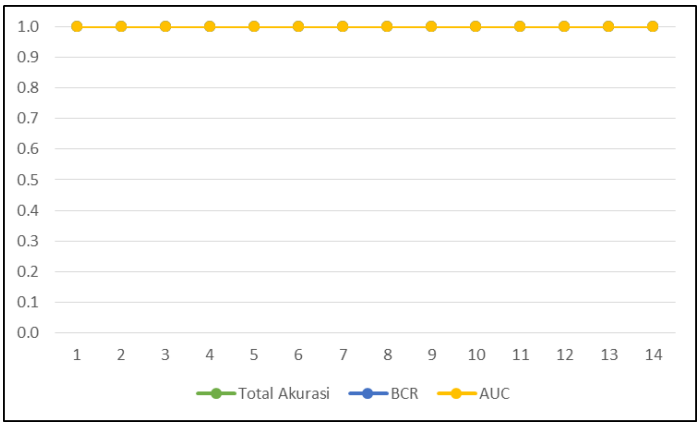
| Partisi | Data *Training* | | | Data *Testing* | | |
|---|---|---|---|---|---|---|
| | Total Akurasi | BCR | AUC | Total Akurasi | BCR | AUC |
| 5 | 1.0000 | 1.0000 | 1.0000 | 0.5400 | 0.5400 | 0.5480 |
| 10 | 1.0000 | 1.0000 | 1.0000 | 0.5900 | 0.5900 | 0.5950 |
| 15 | 1.0000 | 1.0000 | 1.0000 | 0.6500 | 0.6500 | 0.7180 |
| 20 | 1.0000 | 1.0000 | 1.0000 | 0.6200 | 0.6200 | 0.6450 |
| 25 | 1.0000 | 1.0000 | 1.0000 | 0.5900 | 0.5900 | 0.6140 |
| 30 | 1.0000 | 1.0000 | 1.0000 | 0.6150 | 0.6150 | 0.6890 |
| 35 | 1.0000 | 1.0000 | 1.0000 | 0.6000 | 0.6000 | 0.6930 |
| 40 | 1.0000 | 1.0000 | 1.0000 | 0.6400 | 0.6400 | 0.7440 |
| 45 | 1.0000 | 1.0000 | 1.0000 | 0.6150 | 0.6150 | 0.6640 |
| 50 | 1.0000 | 1.0000 | 1.0000 | 0.5950 | 0.5950 | 0.6500 |
| 55 | 1.0000 | 1.0000 | 1.0000 | 0.6650 | 0.6650 | 0.7100 |
| 60 | 1.0000 | 1.0000 | 1.0000 | 0.6450 | 0.6450 | 0.7140 |
| 65 | 1.0000 | 1.0000 | 1.0000 | 0.6400 | 0.6400 | 0.7030 |
| 70 | 1.0000 | 1.0000 | 1.0000 | 0.6600 | 0.6600 | 0.7140 |

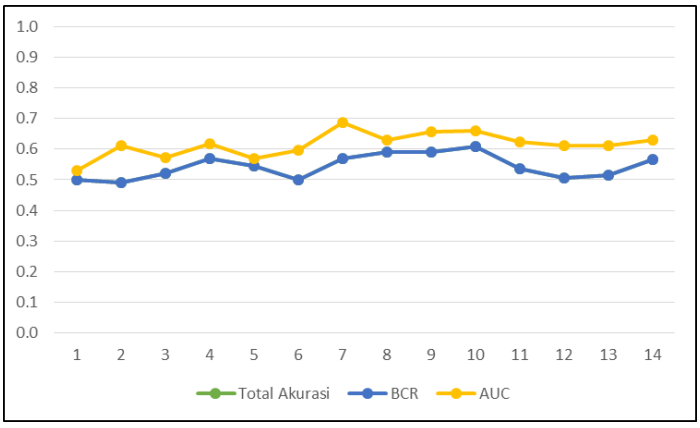


g. Skenario BMV-117

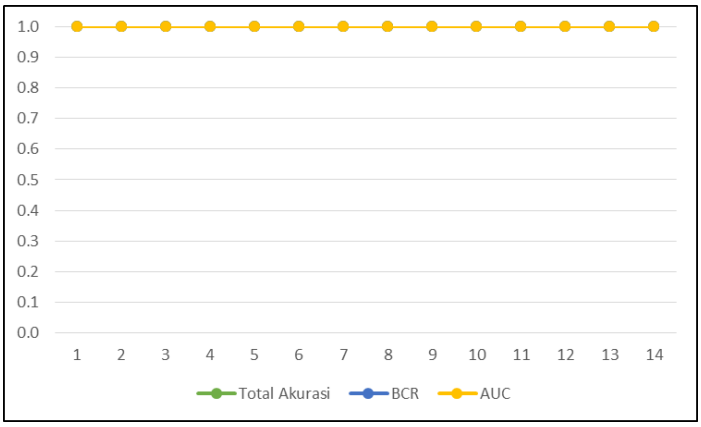
| Partisi | Data *Training* | | | Data *Testing* | | |
|---|---|---|---|---|---|---|
| | Total Akurasi | BCR | AUC | Total Akurasi | BCR | AUC |
| 5 | 1.0000 | 1.0000 | 1.0000 | 0.5000 | 0.5000 | 0.5310 |
| 10 | 1.0000 | 1.0000 | 1.0000 | 0.4900 | 0.4900 | 0.6120 |
| 15 | 1.0000 | 1.0000 | 1.0000 | 0.5200 | 0.5200 | 0.5710 |
| 20 | 1.0000 | 1.0000 | 1.0000 | 0.5700 | 0.5700 | 0.6180 |
| 25 | 1.0000 | 1.0000 | 1.0000 | 0.5450 | 0.5450 | 0.5690 |
| 30 | 1.0000 | 1.0000 | 1.0000 | 0.5000 | 0.5000 | 0.5970 |
| 35 | 1.0000 | 1.0000 | 1.0000 | 0.5700 | 0.5700 | 0.6860 |
| 40 | 1.0000 | 1.0000 | 1.0000 | 0.5900 | 0.5900 | 0.6310 |
| 45 | 1.0000 | 1.0000 | 1.0000 | 0.5900 | 0.5900 | 0.6560 |
| 50 | 1.0000 | 1.0000 | 1.0000 | 0.6100 | 0.6100 | 0.6600 |
| 55 | 1.0000 | 1.0000 | 1.0000 | 0.5350 | 0.5350 | 0.6230 |
| 60 | 1.0000 | 1.0000 | 1.0000 | 0.5050 | 0.5050 | 0.6120 |
| 65 | 1.0000 | 1.0000 | 1.0000 | 0.5150 | 0.5150 | 0.6110 |
| 70 | 1.0000 | 1.0000 | 1.0000 | 0.5650 | 0.5650 | 0.6300 |

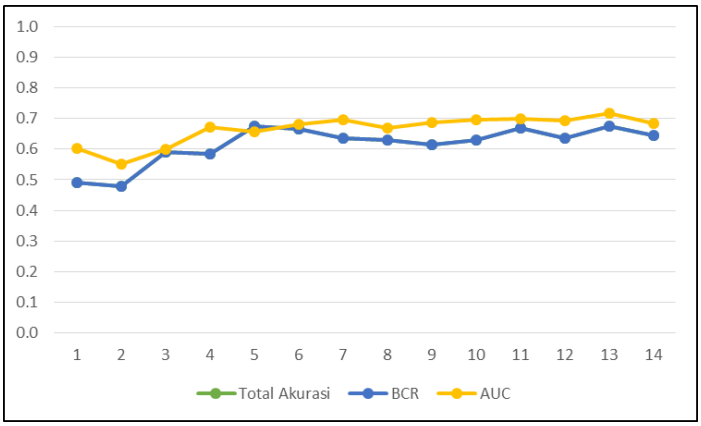


h. Skenario BMV-118

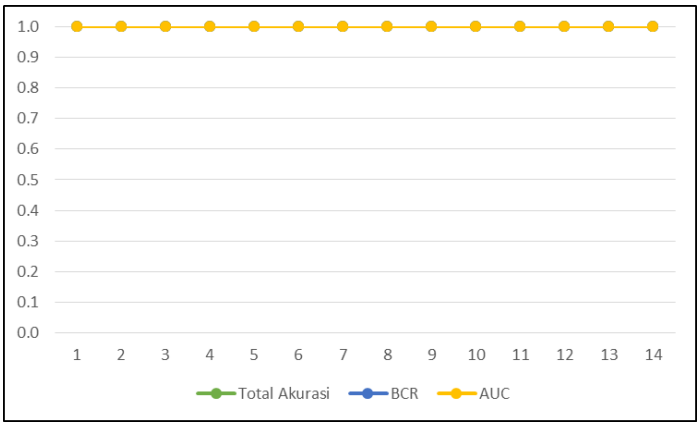
| Partisi | Data *Training* | | | Data *Testing* | | |
|---|---|---|---|---|---|---|
| | Total Akurasi | BCR | AUC | Total Akurasi | BCR | AUC |
| 5 | 1.0000 | 1.0000 | 1.0000 | 0.4900 | 0.4900 | 0.6040 |
| 10 | 1.0000 | 1.0000 | 1.0000 | 0.4800 | 0.4800 | 0.5510 |
| 15 | 1.0000 | 1.0000 | 1.0000 | 0.5900 | 0.5900 | 0.5990 |
| 20 | 1.0000 | 1.0000 | 1.0000 | 0.5850 | 0.5850 | 0.6710 |
| 25 | 1.0000 | 1.0000 | 1.0000 | 0.6750 | 0.6750 | 0.6570 |
| 30 | 1.0000 | 1.0000 | 1.0000 | 0.6650 | 0.6650 | 0.6810 |
| 35 | 1.0000 | 1.0000 | 1.0000 | 0.6350 | 0.6350 | 0.6950 |
| 40 | 1.0000 | 1.0000 | 1.0000 | 0.6300 | 0.6300 | 0.6680 |
| 45 | 1.0000 | 1.0000 | 1.0000 | 0.6150 | 0.6150 | 0.6880 |
| 50 | 1.0000 | 1.0000 | 1.0000 | 0.6300 | 0.6300 | 0.6950 |
| 55 | 1.0000 | 1.0000 | 1.0000 | 0.6700 | 0.6700 | 0.6980 |
| 60 | 1.0000 | 1.0000 | 1.0000 | 0.6350 | 0.6350 | 0.6940 |
| 65 | 1.0000 | 1.0000 | 1.0000 | 0.6750 | 0.6750 | 0.7180 |
| 70 | 1.0000 | 1.0000 | 1.0000 | 0.6450 | 0.6450 | 0.6840 |

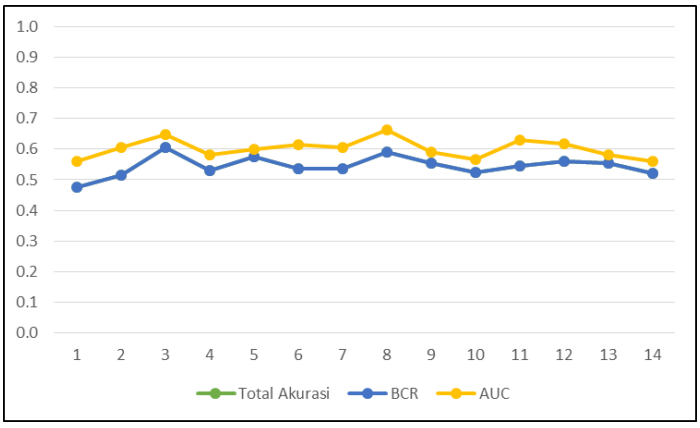


i. Skenario BMV-119

| Partisi | Data *Training* | | | Data *Testing* | | |
|---|---|---|---|---|---|---|
| | Total Akurasi | BCR | AUC | Total Akurasi | BCR | AUC |
| 5 | 1.0000 | 1.0000 | 1.0000 | 0.4750 | 0.4750 | 0.5610 |
| 10 | 1.0000 | 1.0000 | 1.0000 | 0.5150 | 0.5150 | 0.6070 |
| 15 | 1.0000 | 1.0000 | 1.0000 | 0.6050 | 0.6050 | 0.6470 |
| 20 | 1.0000 | 1.0000 | 1.0000 | 0.5300 | 0.5300 | 0.5820 |
| 25 | 1.0000 | 1.0000 | 1.0000 | 0.5750 | 0.5750 | 0.6010 |
| 30 | 1.0000 | 1.0000 | 1.0000 | 0.5350 | 0.5350 | 0.6160 |
| 35 | 1.0000 | 1.0000 | 1.0000 | 0.5350 | 0.5350 | 0.6060 |
| 40 | 1.0000 | 1.0000 | 1.0000 | 0.5900 | 0.5900 | 0.6640 |
| 45 | 1.0000 | 1.0000 | 1.0000 | 0.5550 | 0.5550 | 0.5920 |
| 50 | 1.0000 | 1.0000 | 1.0000 | 0.5250 | 0.5250 | 0.5670 |
| 55 | 1.0000 | 1.0000 | 1.0000 | 0.5450 | 0.5450 | 0.6290 |
| 60 | 1.0000 | 1.0000 | 1.0000 | 0.5600 | 0.5600 | 0.6170 |
| 65 | 1.0000 | 1.0000 | 1.0000 | 0.5550 | 0.5550 | 0.5820 |
| 70 | 1.0000 | 1.0000 | 1.0000 | 0.5200 | 0.5200 | 0.5610 |



**Lampiran 9** Surat Pernyataan Legalitas Data

## SURAT PERNYATAAN

Saya yang bertanda tangan di bawah ini, mahasiswa Departemen Statistika Fakultas Matematika, Komputasi, dan Sains Data, Institut Teknologi Sepuluh Nopember Surabaya:

Nama : T. Dwi Ary Widhianingsih
NRP : 06211650010024

Menyatakan bahwa data yang digunakan dalam Tesis ini merupakan data sekunder yang diambil dari penelitian/~~buku/Tugas Akhir/Thesis/publikasi~~ lainnya, yaitu yang telah dipublikasikan:

Judul : "***Design and Synthesis of 8-Hydroxyquinoline-based Radioprotective Agents***"
Penulis : Shinya Ariyasu, Akiko Sawa, Akinori Morita, Kengo Hanaya, Misato Hoshi , Ippei Takahashi, Bing Wang, Shin Aoki

Surat pernyataan ini dibuat dengan sebenarnya. Apabila terdapat pemalsuan data maka saya siap menerima sanksi sesuai aturan yang berlaku.

Mengetahui
Pembimbing Tesis

Dr.rer.pol. Heri Kuswanto, S.Si, M.Si
NIP 19820326 200312 1 004

Surabaya, 30 Juli 2018

T. Dwi Ary Widhianingsih
NRP 06211650010024

# BIODATA PENULIS

T. Dwi Ary Widhianingsih atau biasa dipanggil TT / Dwi / Dwi Ary, lahir di Jember pada tanggal 20 Mei 1995. Penulis menyelesaikan Sekolah Dasar di SD Negeri Umbulsari 02 tahun 2006, SMP Negeri 1 Umbulsari tahun 2009, SMA Negeri 1 Jember tahun 2012, masuk kuliah di S1 Statistika ITS pada tahun 2012. Dengan berbekal beasiswa *Fresh Graduate*, penulis melanjutkan studi S2 di jurusan yang sama pada tahun 2016.

Selama menempuh studi S2, penulis bekerja paruh waktu di perusahaan *start-up*, Semut Merah Indonesia, yang berada di Jakarta via *remote* atau diskusi online selama satu tahun. Penulis juga ikut berkontribusi dalam beberapa riset Dosen selama dua tahun terakhir. Penulis sangat antusias dengan hal-hal yang berkaitan dengan komputasi statistik dan *machine learning*. Komunikasi lebih lanjut dengan penulis dapat melalui email t.dwiary@outlook.com.

*(Halaman ini sengaja dikosongkan)*